Security for Collaboration in Open, Scientific Computing Environments

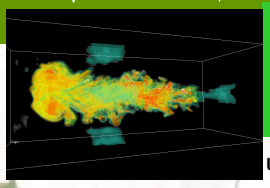
In the future, science will depend on the interaction and interoperation of simulation (computing), data (large-scale archives), scientific instruments, and collaborators at many different institutions. Providing security for these "collaboratory" environments is essential to the well being of this new scientific paradigm.

Security, authentication, and virtual organization policy based access control are required in all aspects of collaboratories.

Users

Collaboratories coordinate people and resources to solve complex problems

Grid services provide uniform access to many diverse resources

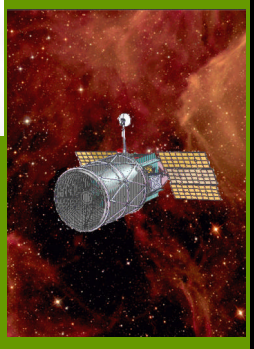


collaboration tools (e.g. shared visualization)

data

computation

scientific instruments



Security for Collaboration in Open, Scientific Computing Environments

A Report of the 4th Joint DOE
Office of Science - Office of Defense Programs
Cybersecurity Workshop

William E. Johnston, Lawrence Berkeley National Laboratory, Convener and Report Editor
(wejohnston@lbl.gov)

Hyatt Regency O'Hare, Ill., Jan. 17-18, 2001

http://www.itg.lbl.gov/DOE_Security_Research/WorkshopIV

Sonsored by
U.S. Dept. of Energy,
Office of Science,
Office of Advanced Scientific Computing Research,
Mathematical, Information, and Computational Sciences Division
(http://www.sc.doe.gov/production/octr/mics)

*Participants and Authors (denoted by *)*

DOE Labs	*Jim S. Rothfuss** *Lawrence Berkeley National Laboratory*
Peter W. Dean *Sandia National Laboratories, Livermore*	*Mary R. Thompson** *Lawrence Berkeley National Laboratory*
*Walter Dykas** *Oak Ridge National Laboratory*	*Cullen Tollbom* *Pacific Northwest National Laboratory*
Douglas E. Engert *Argonne National Laboratory*	*Steve Tuecke** *Argonne National Laboratory*
*Michael Fisk** *Los Alamos National Laboratory*	*Michael O. Vahle* *Sandia National Laboratories, Albuquerque*
*J. D. Fluckiger** *Pacific Northwest National Laboratory*	*John Volmer* *Argonne National Laboratory*
Barry Hess *Sandia National Laboratories, Livermore*	*Ronald Wilkins* *Los Alamos National Laboratory*
*Keith R. Jackson** *Lawrence Berkeley National Laboratory*	**DOE**
*William E. Johnston** *Lawrence Berkeley National Laboratory*	*Thomas Ndousse** *U. S. Dept. of Energy,Office of Science,* *Office of Advanced Scientific Research*
Kyran B. Kemper *Los Alamos National Laboratory*	*Mary Ann Scott* *U. S. Dept. of Energy,Office of Science,* *Office of Advanced Scientific Research*
*Paul Krystosek** *CIAC, Lawrence Livermore National Laboratory*	**Universities**
Bob Lukens *Jefferson Laboratory*	*Dennis Gannon** *Computer Science Dept., Indiana University*
Bob Mahan *Pacific Northwest National Laboratory*	*Sara Matzner* *Applied Research Laboratories,* *The University of Texas at Austin*
W. Frank Mason *Sandia National Laboratories, Albuquerque*	*Barton Miller** *Computer Science Dept.,* *University of Wisconsin - Madison*
*Sandy Merola** *Lawrence Berkeley National Laboratory*	*Clifford Neuman* *Information Sciences Institute,* *University of Southern California*
*James Rome** *Oak Ridge National Laboratory*	*Thomas Shields* *CERIAS, Purdue University*

Introduction

Modern science increasingly depends on experiments that involve large, specialized instruments, management of huge amounts of data, intensive computing for both data analysis and simulation, and human collaboration, all of which are scattered among many institutions. *Collaboratories* is the name given to the networked communication and data frameworks that connect people, computers, and instruments to make large-scale science possible. Part I of this report describes the importance of collaboratories and open research environments, and Section 1 illustrates how these large-scale systems are at the heart of the Department of Energy's (DOE's) science mission. In particular it looks at DOE's Office of Science programs, their scientific collaborations, instruments, and computing and data resources, as well as the Office of Science's supercomputer facility, the National Energy Research Scientific Computing Center (NERSC).

Almost all collaboratories involve multiple institutions – DOE labs, universities, industry, other agency labs and systems. This means that most collaboratories operate in computing environments that are inherently open, involving multiple administrative domains without common security models, even without common threat models. Yet security – access control, confidentiality, and uninterrupted service – is essential for collaboratories to function. The individual participants must agree on what needs to be kept private, the computing systems must not be disrupted by hackers, and the networked instruments must be protected from any sort of cyber tampering.

The unique challenges of providing security in an open, scientific environment are introduced in Section 2 of this report. Many standard protection measures that work well in the commercial or military sectors can actually have severe detrimental effects in the open research environment. Security policies for collaborative scientific communities must take into account the nature of those communities and how they work. For example, collaboration cannot flourish in a fortress mentality; for scientists, protection of service is frequently more important (and more difficult) than protection of information. DOE and its laboratory and university community must establish cyber-security policies that enable and protect the success of the DOE science mission, and these policies will differ from those developed by organizations with different missions. Further, managing the human factors that affect security will be as important as implementing the latest security technologies.

Part II of this report (Sections 3 and 4) looks at the open scientific computing environment five years from now and the potential security threats we will face. The emergence of computational and data Grids – standardized middleware for managing the distributed, large-scale computing and data resources of science and engineering is described in Section 3. There will be many advances in the computing and telecommunications milieu that we could not protect with today's security tools, even assuming that attackers and their tools did not change, which will certainly not be the case. The ways that we use and access computing will change, the numbers of researchers and students using collaboratory environments will increase, and the sophistication of hackers will increase substantially. This is considered in Section 4.

Therefore, the criticality of security *together with* easy access for collaboratory systems will increase dramatically over the next five years. We must become more sophisticated in how we provide security in an environment that will be increasingly open in terms of the diversity of its population and in its use of open infrastructure. We must also provide security in ways that do

not interfere with easy access to the many different services needed to build and use large collaboratories, and do not degrade the high performance of networks and systems needed for scientific productivity. Security, easy access, and high performance must all be maintained. If any one of these three elements fails, the collaboratory systems will fail.

Part III addresses the state of distributed applications security and identifies security issues that require further research and development if they are to be addressed successfully. Section 5 discusses collaboratory security issues such as authentication and authorization, perimeter protection, ease of use, protection and performance tradeoffs, metrics and analytical models, and others. Section 6 explores collaboration domains and enclaves, i.e., cross-organizational resources that share a common security policy. Section 7 discusses issues of code safety, including both mobile code and code reliability. And Section 8 describes the need for a cyber-security science that goes beyond current ad hoc approaches to a disciplined methodology for analyzing and modelling cyber-security scenarios as well as validating security techniques and systems against benchmark metrics and specific requirements.

Part IV presents the conclusions of this report. Because DOE's mission gives it a leadership role in building and using large-scale collaboratory environments, DOE must also take a leadership role in ensuring their security, accessibility, and high performance. This leadership will include sponsoring research into the unique security issues facing open collaboratories, and development of appropriate solutions.

Part I: Importance of Collaboratories and Open Environments

1 Collaboration, Collaboratories, and the DOE Mission

As science tackles more and more subtle and complex problems, the scale of the scientific endeavor increases in every dimension. For example, High Energy Physics' attempt to reveal the fundamental nature of matter has reached the point where experiments cost a billion dollars, involve hundreds or thousands of collaborators from around the world, and last a decade. Such science is not possible without major involvement of advanced computing, data, and network technology that can serve all of the collaborators.

In this scientific environment scientists can no longer work in isolated laboratories. The problems being studied require experts from several fields to combine their talents. The instruments used to gather scientific data are expensive and difficult to build, so there are only a few of each type in the world. The Internet connectivity and the powerful desktop workstations available today provide the potential to create collaboration tools and allow organizationally and geographically dispersed scientists to collaborate and access instruments remotely.

The goal of *Collaboratories* is to provide a location independent laboratory where scientists can collaborate. Collaboratory technologies include videoconferencing over the Internet, remote access to electronic notebooks, shared visualization, security infrastructure, floor control, and session management. A collaboratory presents the users with new ways of thinking, working with others, and doing tasks. For instance, with an electronic notebook, data can be automatically recorded, potentially from multiple sites, to the electronic notebook as it is taken. Meetings can be held without needing a conference room or even needing to meet at the same site. A group that is collaborating but is separated geographically can use virtual office spaces to bring the group together into a common space.

Additionally, in order to support this shift in science, the computing and data handling infrastructure that is essential to most of modern science must change to support the complexity of the multi-disciplinary and multi-step simulations and data analysis found in large-scale science and engineering. Computing and data Grids are software infrastructure that facilitates solving large-scale problems by providing the mechanisms to access, aggregate, and manage the computer network based infrastructure of science. This infrastructure includes computing systems, data archive systems, scientific instruments, and human collaborations (through the collaboratory technology mentioned above).

In this section we will look at the motivation, impact, and evolution of several emerging collaboratory and Grid environments. We will also characterize the computing and networking of these collaboratories that provide the challenges for cyber security. In several cases, the use of emerging Grid technology illustrates the direction of the evolution.

1.1 High Energy and Nuclear Physics

One of the most basic science missions of the Department of Energy is the understanding of matter and energy in the universe. The High Energy Physics program has as its primary focus the constituents of matter and the fundamental forces that govern their interactions. The Nuclear Physics program focuses on the structure of matter and the nuclear processes at work in the universe. The major high-energy particle and nuclear physics experiments of the next twenty years will break new ground in our understanding of the fundamental interactions and symmetries governing the nature of matter and space-time [1]. Since the realization of these groundbreaking results involves the extraction of small or subtle new physics signals from large and potentially overwhelming backgrounds, realizing the scientific wealth of these experiments presents new problems in data access, processing, and distribution. Furthermore, these problems are being faced by ever growing collaborations of researchers spanning national and international networks, on a scale unprecedented in the history of science. There is a growing realization that, without the collaborative infrastructure that makes it possible for physicists in all world regions

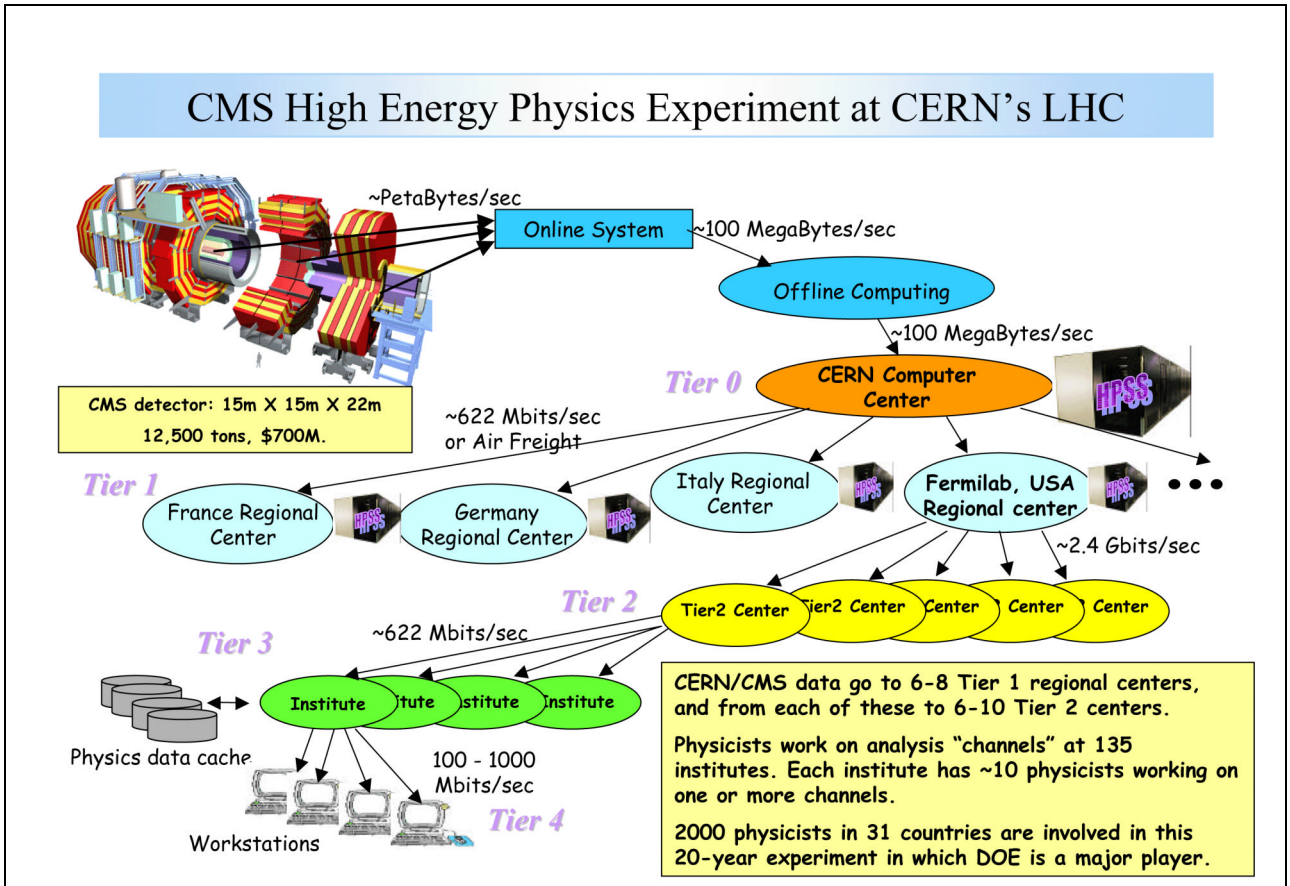


Figure 1. DOE's High Energy Physics program – for which CMS is a typical example – epitomizes the need for collaboratories supported by Grid computing infrastructure in order to enable new directions in scientific research and discovery.
Figure courtesy H. Newman, Cal Tech and CERN.

to contribute effectively to the analysis and the physics results, these research efforts will not succeed.

The management and analysis of the extremely large quantities of data produced by leading high energy and nuclear physics experiments represents an unprecedented information technology challenge (Figure 1-1). These efforts must provide rapid, transparent access to experiment data samples and subsets drawn from massive data stores, growing from hundreds of terabytes in 2000 to petabytes by 2005, and ultimately to 100 petabytes (100 million gigabytes) by 2010. There is a broad realization within these communities that the computational and storage resources needed for data management and analysis cannot realistically be gathered at a single location, and that future computational environments must hence be distributed collections of storage systems and compute farms, i.e., "Data Grids," that are operated in a coordinated fashion [2], [3].

Over the last few years this concept has evolved into that of a data-intensive, hierarchical "Grid" of national and regional centers linked to the principal center at the experimental site, and to local computing resources. There is considerable synergy between these developments and work in computational and data Grids; further, in other fields (e.g. [4]), Grid technology is emerging as the infrastructure that supports the construction of these types of collaboratories.

For example, over the past two years, the DOE-funded Particle Physics Data Grid (PPDG) project [5] has explored the use of Grid technologies for distributed management and analysis of data from major experiments such as BaBar, D0, CMS, and ATLAS. PPDG participants have demonstrated combined management and high-speed transfers of physics data between DOE labs and universities using a combination of Grid services, technologies such as Condor for distributed data analysis, and replica catalogs for data management. These experiments have been highly successful and have resulted in a solid understanding and extensive community concerning Data Grid requirements and architecture. Based on this understanding and consensus, the PPDG-2 project now plans to establish production Data Grids for a range of DOE-funded physics experiments.

The adoption of Grid concepts by this community represents an important endorsement of the technology, but also introduces significant challenges. Specifically, they now face the need to deploy production services for authorization, resource discovery, resource access, etc. In the absence of a central coordinating site, they will inevitably be forced to create their own core security services and run their own servers such as directory servers. Unless great care is taken, there is no assurance of interoperability with other DOE Grid and collaboratory components.

1.2 Chemistry

A large fraction of all scientific computing cycles are devoted to computational chemistry, and computational chemists often use high performance computers at distant institutions to perform their work. Although there are many computational chemistry codes, the vast majority of computing cycles are used by a small number of closely related codes for electronic structure and molecular dynamics computations (e.g., Gaussian, Gamess-US/UK, NWChem, Amber, Charmm, Gromos). Many of these codes have been ported to a wide variety of computing platforms. For any given study, the same code may be run on many systems, ranging from a desktop computer to distant terascale systems. Therefore, a small number of Grid implementations of chemistry

applications will leverage a large number of compute cycles, and serve as models for many other similar codes.

Managing the increasingly complex array of computations and resources has traditionally been a problem mediated by handcrafted scripts and paper records. Problem Solving Environments (PSEs) – the user interface for collaboratories and Grids – provide interactive tools for integrating and managing these computations, supporting the discovery process, and managing databases of computational results. An example of a PSE is Pacific Northwest National Laboratory's (PNNL's) ECCE – the Extensible Computational Chemistry Environment [6] – which manages calculations for both Gaussian and NWChem. (Gaussian is arguably the world's most popular electronic structure code, and NWChem is used at hundreds of sites for large, scalable computations.) However, the impact of these PSEs is limited without standard tools and interfaces for locating resources, authorizing and authenticating access, transferring data, launching jobs, etc. ECCE uses the Globus [7] implementation of Grid technology to provide a standard way to access remote resources and provide standard security services across sites. The Globus Grid approach is the only one currently that promises to have the flexibility needed for present and future generations of computational chemistry codes and their associated collaborative problem-solving environments. The availability of a persistent Grid is crucial to providing a stable, secure, and open environment that is required for the community developing chemistry codes and their PSEs.

1.3 Materials

The Materials Micro Characterization Collaboratory (MMC) [8], a joint effort between Oak Ridge National Laboratory (ORNL), Argonne National Laboratory (ANL), and Lawrence Berkeley National Laboratory (LBNL), was one of two collaboratory pilots funded by the DOE2000 Program . The primary focus of the MMC was remote instrument control (microscopes and beamline experiments). Over the past four years this project has put into place a collaboratory framework that caters to the needs of the microscopy community. Their framework is now used routinely for remote access to lab resources by industry and other DOE researchers. The Materials MicroCharacterization Collaboratory focuses on the scientists as users in an interactive electronic laboratory. The collaboratory provides opportunities for creative scientists with varied yet complementary expertise to come together in an environment designed to allow rapid and dynamic interactions to flow unencumbered by the limits of time and distance. In it they have developed a virtual environment equipped with state-of-the-art research capabilities (consisting of both scientists and instrumentation) for microcharacterization and materials research. The MMC has also pushed the frontiers of electronic lab notebooks as a means to collaborate and share research in a production environment.

Much of the underlying infrastructure for communication, security, and remote instrument access was created specifically for the MMC project. Although the MMC collaboratory pilot has been successful, a common framework for building collaboratories is needed to move to a production infrastructure. The DOE Science Grid [9] will provide a framework for the common collaboratory infrastructure to help avoid replication of collaboratory framework efforts across subsequent collaboratories and to provide standardized remote access and security.

1.4 Climate

The need to evaluate climate change scenarios as a basis for federal policy planning makes climate modeling a mission-critical application area for DOE. DOE climate modeling work seeks to address this need through the creation of an advanced climate simulation program that will accelerate the execution of climate models one hundred-fold by 2005 relative to the execution rate of today. High-resolution, long-duration simulations performed with these models will produce tens of petabytes of output. However, to be useful, this output in turn must be made available to global change impacts researchers nationwide – at national laboratories, universities, other research laboratories, and other institutions. To this end, DOE researchers are working to create a collaboratory called the *Earth System Grid* (ESG): a virtual collaborative environment that links distributed centers, users, models, and data. This ESG will provide scientists with virtual proximity to the distributed data and resources that they require to perform their research through seamless and high-performance access to data and compute resources.

The ESG project, which involves researchers at ANL and LBNL as well as the National Center for Atmospheric Research (NCAR), the Information Sciences Institute at the University of Southern California (USC/ISI), and Lawrence Livermore National Laboratory (LLNL), has created data-replica management and data transfer tools, and integrated these tools into climate model data analysis systems to enable automatic selection of the "best" copy of data. Experience with the use of these tools has emphasized the importance of persistent Grid infrastructure services for production use, so that ESG users can authenticate themselves, determine availability of Grid resources, and then access and/or transfer required datasets. For example, much of the climate data of current interest to ESG users resides at DOE's National Energy Research Scientific Computing Center (NERSC), and secure integration of NERSC facilities with other resources of the climate community will be essential for the success of this work.

1.5 Supernova Cosmology

Over the past several years, astronomers and astrophysicists have been conducting in-depth sky searches with the goal of identifying certain reference types of supernova in their earliest evolutionary stages and then, during the two to four weeks of their most "explosive" activity, measuring their changing magnitude and spectra. These "standard candles," as they are called by the astronomers, are supernova that can be used to directly measure various cosmological properties. (See [10] and [11].) These early experiments have demonstrated that the expansion of the universe is accelerating, apparently driven by an unknown new force that overwhelms the force of gravity, contrary to existing models where gravity would cause the universe expansion to slow. The discovery of this new force – now called dark energy – is a stunning discovery and was named the "breakthrough of the year" by Science Magazine in December,1998.

These experiments have been daunting tasks in terms of both the number and volume of observations required. The early successes have driven the expansion of these searches in terms of both sky area and apparent magnitude observed. The search program currently under development at LBNL, the Supernova Factor (http://snfactory.lbl.gov), is an earth-based observation program utilizing observational instruments at Haleakala and Mauna Kea, Hawaii and Mt. Palomar, California. When fully implemented, this search program will also utilize instruments at observatories in Chile and the Canary Islands. This program will also serve as a

development testbed for the next generation search program, the space-based Supernova Acceleration Probe (SNAP). The Supernova Acceleration Probe is a satellite-based supernova search program combining an optical field imager, near infrared imager and spectrometer in a single, dedicated spacecraft (see http://snap.lbl.gov).

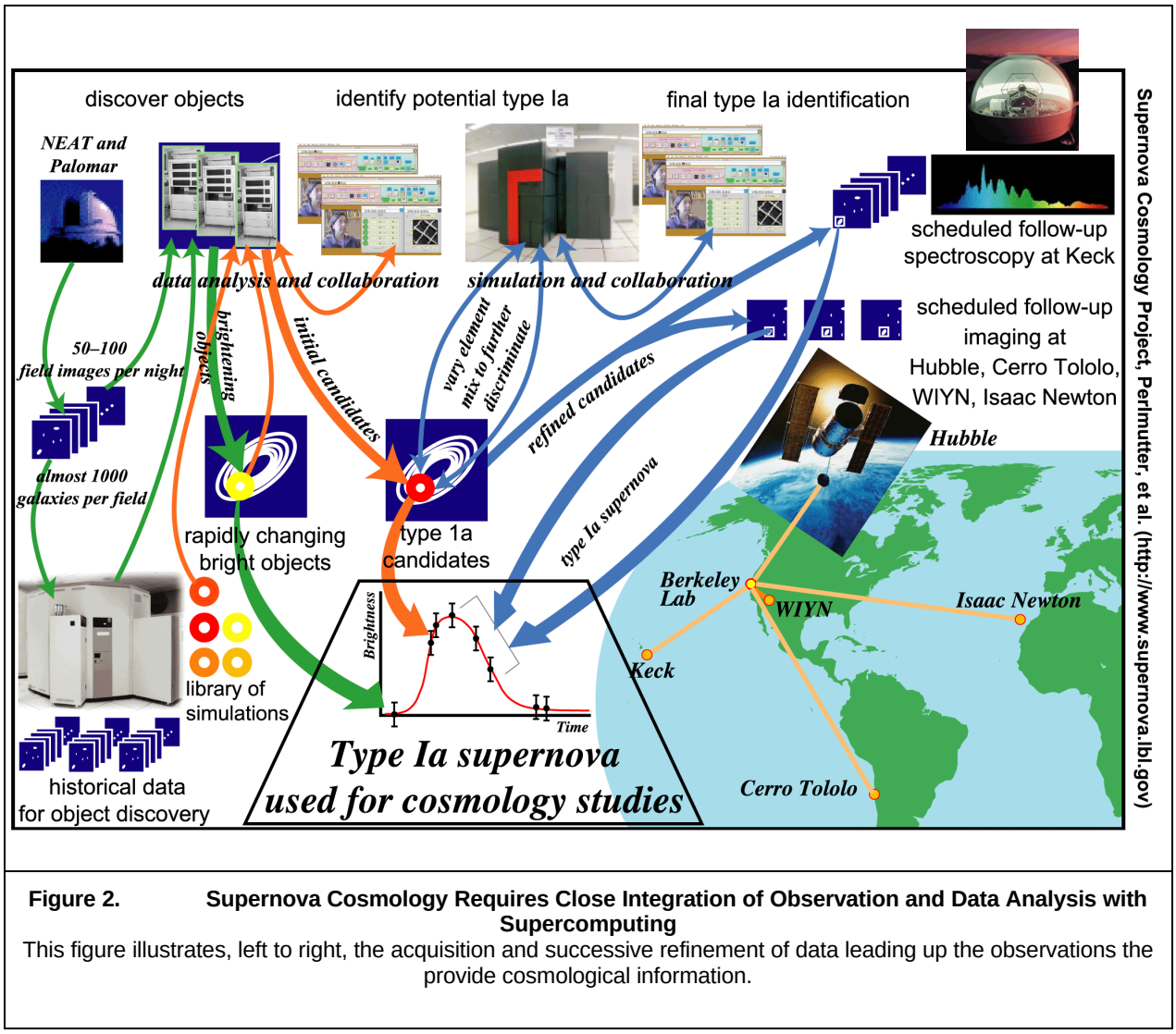


Figure 2. Supernova Cosmology Requires Close Integration of Observation and Data Analysis with Supercomputing

This figure illustrates, left to right, the acquisition and successive refinement of data leading up the observations the provide cosmological information.

This new approach to cosmology – only possible because of the availability of large-scale computing and data storage facilities at the DOE NERSC facility and the corresponding National Science Foundation (NSF) supercomputer centers – is called "observational cosmology.

The evolution from proof-of-principle to full scale supernova search has unveiled new operational issues for these research programs that we feel are characteristic of how modern science is evolving under the influence of vastly increased distributed computing and data handling capabilities. The first of these is the sheer scale of the computing and data handling task involved. Raw, uncorrected sky images must be transferred nightly from remote observatories to central computing facilities, NERSC [12] in this case. Here, these images undergo extensive computational calibration and correction to eliminate sky tracking errors as

well as instrumentation and atmospheric effects. The resulting images must then be compared to recent baseline sky catalogs in order to eliminate asteroids and man-made satellite tracks. Only then can automated search algorithms look for increases in stellar magnitude that may indicate the onset of supernova activity. Fifty plus Gigabytes in some 500 files need to be shepherded through this process of data transfer, computation and archiving on a daily basis for the five to ten years duration of the search effort. The script and operator based automation used during early sky search programs simply will not scale to the levels of performance and reliability required by these new searches.

Secondly, the amazing experimental results obtained thus far have promoted strong programmatic ties between cosmologists involved in modeling stellar behavior through simulations and those engaged in direct observation. Simulation teams are now engaged in ambitious efforts to develop new models that provide full 3D simulation of both the hydrodynamic and radiative transfer aspects of supernovae that can predict, based on the parameters of the exploding star, the spectra during supernova. Since the development of accurate models requires a detailed comparison with observed supernova data, data from the Supernova Factory is of critical importance in the successful development of these models. Although the initial motivation is the improvement of current computational models through direct and frequent comparison to observations, ultimately the goal is to use closely-coupled observation/simulation efforts to filter out supernova candidates that are not the reference types useful for cosmology. As both the number of discovered supernovae and the demands for scarce, shared observational instruments increase, the ability to successfully filter unwanted supernovae out of the observational program becomes increasingly important. This is accomplished by using the initial observation to establish the parameters for the simulations which, in turn, predict the observed spectra in order to determine the exact type of the supernova.

When this determination of type results in identifying a "standard candle" (type 1a) supernova, this information must be immediately conveyed to one of the large instruments such as Keck, Palomar, or Hubble, in order to observe the spectra throughout the short (weeks long) life of the supernova. (See Figure 2.) This is the information that permits cosmological inference. This establishes a cycle of coarse observation – simulation – detailed spectra observation that is time constrained by the fact that the useful spectrum observation period is only for a few weeks following discovery. See Figure 3.

This work is inherently collaborative, and real-time collaboration is essential for its success. The scientists participating in the simulation development and in the measurements are themselves widely distributed, and furthermore, the telescopes, instruments and computers used in the search

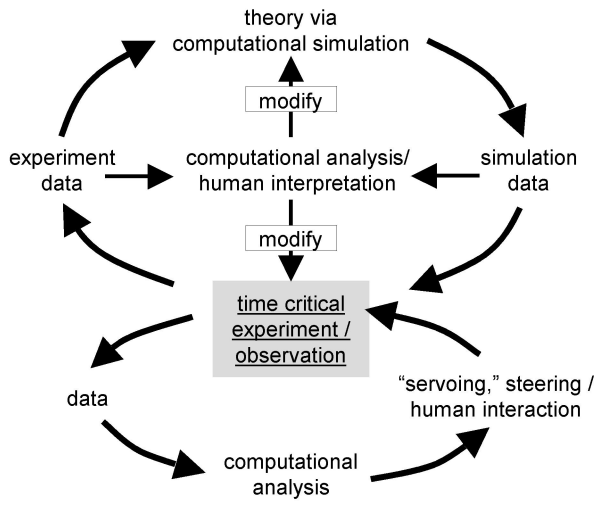


Figure 3. The workflow characterizing the supernova search project

are distributed throughout the world. Effective interactions with staff at remote observatories will play an increasingly key role in successful daily operations. During both sky survey and follow-up observations, it is often necessary to interact with observatory staff to adjust instrument settings, inquire about current sky conditions, and quickly schedule repeat observations. Telephone and email are not effective for these interactions, and integrated collaboration tools become necessary.

Security and authorization acquire significant importance when developing mechanisms that allow collaborators throughout the world to monitor and control daily analysis and archiving efforts. Success will depend on collaborating scientists being able to manage data processing and storage and to integrate advanced supernova simulation into the real-time control of the experiments. The ability to perform real-time control will allow collaborating scientists in one part of the world to look at results and change viewing plans in another part, thus taking advantage of the different time zones across the collaboration – an important aspect when observation can only be done for a few hours each night.

As we see, the computing requirements of both the present Supernova Factory and future supernova search programs cannot be simply stated in terms of compute cycles and storage capacities. These research programs anticipate a robust, highly distributed, easily managed computing environment that will support their long-lived research program. In short, they need a stable and scalable software environment that provides consistent services and interfaces to their applications as underlying systems continue to evolve and improve. They are currently in the process of adopting the Grid to provide the foundation for such an environment. However, as noted earlier, the underlying structure is, by itself, insufficient to support scientific programs such as the Supernova Factory. Computing and instrumentation resources need to be integrated into Grid Information Services; security information and certificate services need to be federated with those of other collaborating institutions; and integrated workflow management tools need to span all participating computing resources, both local and remote. Implementation, deployment, and support of this environment is well outside the scope and capability of these research programs but is necessary to their continued expansion.

1.6 Efficient Diesel Engine Design

Due to their high efficiency and reliability, diesel engines are the dominant power source for heavy-duty trucks and buses in the U.S. Heavy-duty diesel thermal efficiency is about 45%, versus 30% for production gasoline engines. Even higher efficiencies are possible, and the implementation of small-bore diesel engines promises to greatly improve the fuel efficiency of autos and the rapidly growing light truck market. The development of chemically predictive model-based engine design tools is a critical part of the DOE's strategy to meet the efficiency and emission requirements over the next decade. However, combining the modeling, the engineering design, and the validating experiments presents unique challenges.

As seen in Figure 4, the overall process is a complex task that requires a close interplay among experimentation, development of physio-chemical submodels, development of numerical codes for engine simulation, model validation and formulation of new modeling concepts, the development of new engine concepts and hardware leading to new experiments. Ultimately, this process is envisioned to enable the simulation-based design that will revolutionize the production of new diesel engine technologies. Progress in this already difficult task is further complicated

by the distributed, multi-institutional, and interdisciplinary nature of the required collaboration, and the computing and data handling systems, among the geographically distributed industry, national laboratory, and university partners.

The Diesel Combustion Collaboratory (DCC) [13] is a research collaboratory pilot project that includes collaborators distributed among DOE national laboratories (Sandia (SNL), Lawrence Berkeley (LBNL), Los Alamos (LANL), and Lawrence Livermore (LLNL) national laboratories), the University of Wisconsin, and industry partners (Caterpillar, Cummins Engine, and Detroit Diesel). The DCC itself is a problem-solving environment for combustion researchers. The purpose of the DCC is to make doing the science, engineering, and information exchange for the combustion researchers more efficient. High-speed networking, computer

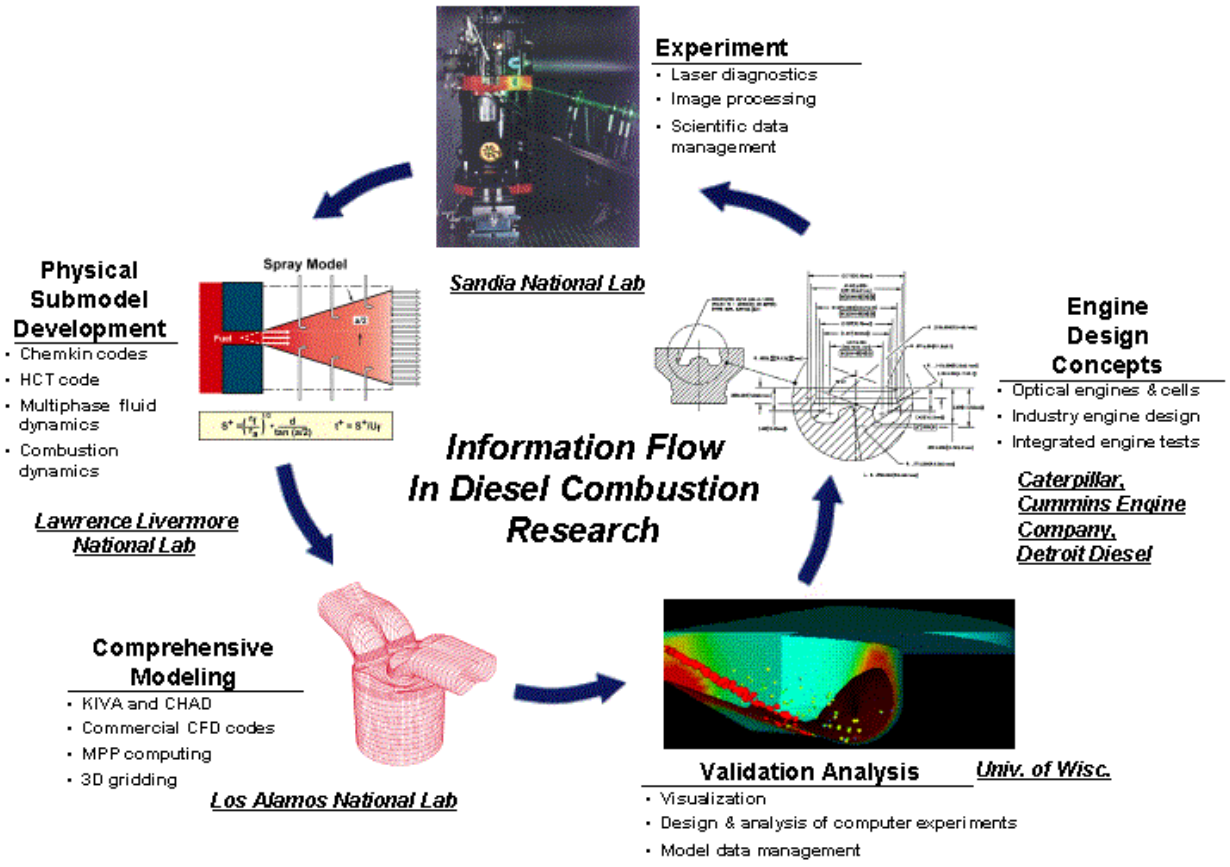


Figure 4. **Design of complex systems require the interaction of many disciplines, and the interaction of their computing, data, and experiments.**

security, distributed computing technology, data visualization, and collaborative tools are essential parts of the underlying infrastructure for the DCC.

Each institution in the collaboration offers a subset of the required interdependent capabilities that must be integrated into a collaboratory for the overall success of the program. The goal of

the Diesel Combustion Collaboratory was to make doing the science, engineering, and information exchange for the research partners much more efficient. The inclusion of competing groups (the industrial partners) in the collaboration brought with it very strong requirements for data security. The data access rights had to be easily controllable at a very fine grain while ensuring that the privacy requirements were adhered to, even when the data was stored on third-party systems that are not the owner of the data. The DCC enables researchers to tackle new problems with their existing methods simply because the techniques are easier to use in a collaboratory environment, and it provides access to new methods through collaborations that were previously inaccessible because of geographic or expertise limitations.

Directory services (for resource management), Grids providing standard middleware, the persistent infrastructure of the Grid services, will play a critically important role in recreating this sort of environment without duplicating development efforts.

2 Security Issues for Open, Networked Scientific Environments

As illustrated in the previous section, almost all collaboratories involve multiple institutions – DOE labs, universities, other agency labs and systems, etc. This means that most collaboratories operate in inherently "open" networked environments. For the purposes of this discussion, one critical aspect of this is that open, networked, scientific computing environments are almost always heterogeneous: They involve multiple administrative domains without common security models, frequently without even common threat models. Open research environments typically are more concerned about protecting systems and services than about holding information confidential. However, collaborations with industry often require that confidentiality is also enforced. In the Diesel Combustion Collaboratory, for example, different industrial partners share resources and some data, but other data must be kept company confidential.

Scientists are usually motivated to join an open scientific research environment because it offers resources critical to solving their problems faster and more efficiently, and in an increasing number of cases, it is the only way to solve their problem (as the high energy physics and observation cosmology examples demonstrate). Ownership and protection of information is strictly a function of the competitive nature of scientific research, and government agency and even university policies in this area are likely to be viewed by all participants as a nuisance at best, and an impediment to progress at worst.

Yet security is essential for collaboratories to function: The individual participants will agree that some data must be kept private, that computing systems cannot be disrupted by hackers, that networked instruments and data must be protected from any sort of cyber tampering, etc.

Rational security policy imposed on open scientific communities must take into account the nature of those communities and how they work. The overly restrictive policies that result from applying watered-down versions of security policy from different environments – e.g., ones involving national security issues – will virtually always have a detrimental impact on the collaboratories of open scientific communities. As illustrated in the examples above, most large-scale scientific communities that address large and complex problems are multinational, thus making creation and application of global security policies even more problematic.

2.1 Principles of Protecting the Open Research Environment

Jim S. Rothfuss, Computer Protection Program Manager, Lawrence Berkeley National Laboratory

The assumption is often made that a single list of cyber-security rules can apply to all environments and that variations can be accounted for by varying the degree of rigor with which the rules are applied. In practice, this assumption is false. In the world of physical security, the rules that an army uses to protect a military base are radically different than the rules a protective service force uses to protect a university campus, even though both fall under the heading of "security." In the same way, the cyber-security "rules" to protect a computer used for classified research are quite different than those used to protect a computer used for open, public research. Following are some observations of what sets an open research environment apart from a military or commercial environment.

2.1.1 What Needs to Be Protected?

In general, there are two attributes of computer use that need to be protected: the information that is stored on a computer and the ability of the computer to function, or "do its job." In the first case, the primary concern is that the data are not exposed. In the second case, the concern is that the computing resource is readily available. Historically, cyber security has focused primarily on information protection. An example is nuclear weapons design, where the exposure of weapons data is treated with intense concern, but the computers doing the design work can be shut down for hours, perhaps even days, with little lasting damage. In contrast, recent commercial endeavors such as Web portals base their business model on exposing as much data as possible, but minutes of down time may cost millions of dollars. Ironically, information protection and service protection are often at odds. Getting the wrong mix will almost always result in detrimental effects. In the open research environment, information needs to be protected, but not with the rigor of national security rules. For computers, data stores, and instruments, downtime of minutes might be tolerated, but hours or days down can be quite detrimental to the research activity, if, for example, an experiment that depends on these systems is in progress. Since the word *open* implies public and *research* (in this context) implies using computers, the balance tends to fall on the side of keeping the computers in use rather than protecting information. A great deal more attention needs to be placed on protection of service than it has been given in the past, and in some ways this is a more difficult problem than protection of information.

2.1.2 Open by Default

The word *security* often brings to mind a picture of a big lock. Every door is locked and no one goes in or out with out the proper key. This is the *restrict by default, allow only as necessary* philosophy. In the real world, using locks is more the exception than the rule. Stores, college campuses, and cities all work on the principle that the public is free to enter, restricting only those areas that are off limits. Much of cyber space is the same. Commercial businesses are finding that a *restrict everything* firewall policy also restricts interactions with the public and other businesses. In short, it restricts close interactions with their sources of profit. In a similar fashion, too many restrictions placed on a researcher's ability to use computers will simply deter the researcher from doing research. Certainly some restrictions are necessary, but the goal should be to minimize the restrictions and maximize the researcher's freedom to explore.

2.1.3 Networks Promote Collaboration

In 1931, E.O. Lawrence took the visionary approach of encouraging collaboration among various scientific and engineering disciplines. The results of that approach are DOE's national laboratories and "big science" solutions to big problems. Computer networks offer the hope of enormous collaborations, not only between localized disciplines, but also between laboratories and research facilities throughout the world. This offers the prospect of solving enormous problems. This spirit of collaboration cannot exist in a "fortress mentality" of isolating each facility into its own separate slice of cyber space. A tremendous challenge exists to protect collaborative organizations intermingled throughout the Internet.

2.1.4 Fluid Change Is the Norm, Not the Exception

Computer systems in a commercial environment are often required to repeat the same tasks or transactions for years. Steady state is a valued attribute. In fact, a business may go to great lengths to ensure rigid configuration control at the cost of deterring technological progress. Research, by definition, is not steady state. Rapid prototype and ad hoc experimentation are what often lead to technological breakthroughs. Ways must be found in research environments to allow fluid change in the cyber environment within the bounds of acceptable risk and without the burden of undue configuration control overhead.

2.1.5 Research Is the Ability to Create

If an operator for a computerized weapons system is needed, it is entirely possible that a new, 18-year-old military recruit can learn the job with a few weeks of intensive training. Similarly, in a business, system managers may be hired to tend an important transaction machine. In both of these examples, the computer is the focal component that meets the strategic goals. The operator or system manager job exists to support the computer.

The creation and development of innovative ideas are what sustain a research environment. Computers cannot create ideas. People create ideas, using computers as tools in the process. In the implementation of cyber security in a research environment, it must be remembered that the goal is to maintain the researcher's ability to create. Protecting computers is only important in the context of how the computer is used to support the scientific discovery process.

2.1.6 The Open Research Environment Is Different

In an open research environment, it is important to understand the differences from a commercial or military environment lest the mission be undermined.

- Emphasis on information protection may fatally undermine service requirements.
- A highly restricting environment may be safe, but it is not effective.
- Cyber fortresses severely inhibit network-based collaboration.
- In research, the capabilities of computers need to be pushed to their limit.
- Ideas are the foundation of research, and computers do not create ideas, people do.

Many standard protection measures that work well in the commercial or military sectors can actually have severe detrimental effects in the open research environment. Much work remains to characterize this environment and to develop optimum tools and processes for its protection.

2.2 Collaboratory Needs *versus* Security Policy and Infrastructure

James A. Rome, Oak Ridge National Laboratory

2.2.1 Facilities Must Be Accessible but Secure

DOE facility users come from industry, universities, and other DOE labs, and may or may not be U.S. citizens. Users may come to a lab for one short session, and later interact over the Internet, or they may never physically appear at a DOE lab. Although DOE security policies are aimed primarily at protecting DOE and its facilities, the user might also have strong requirements for the protection of his proprietary data, processes, or techniques. This diversity poses a challenge to a "one size fits all" security access policy.

On the other hand, security is necessary, and expensive and delicate facilities should be protected by the best available security technology, e.g., firewalls, encrypted traffic, strong authentication and authorization.

The boundaries for security are often not clear. A facility at one lab might be owned by another lab (e.g., the X14 ORNL beamline at BNL). Large-scale problems require distributed computing and storage, and will probably use resources at many DOE sites. There is a strong motivation to have uniform ways of satisfying the multi-site security requirements if the number of individual site requirements is not prohibitive.

2.2.2 Security Barriers Raise Difficulties

DOE is encouraging each lab to wrap itself in firewalls, to create enclaves, and to "know" its users. The question then arises, "Who should be allowed into these enclaves?" Should the authentication instruments (e.g., a Public Key Infrastructure [PKI] certificate) of one lab be accepted at others? Do these instruments contain sufficient information to allow the other lab to make an intelligent access decision? If each lab issues its own authentication tool or device, is it reasonable to say "come get your crypto card at our office during business hours" for remote users?

DOE fielded a very successful cross-realm (multi-lab) Kerberos/DCE authentication project. However, the end result was not always attractive because of policy issues, rather than technology. There are fundamental issues about the size of a security realm that are hard to overcome. In the *King and I*, the King had a song about alliances – if they are too small, they won't help you and if they are too big, you can't trust them. Similar considerations apply to security domains. If they are too small, they cannot afford to spend the resources necessary to manage their security so that you can trust them, and if they are too big, it is hard to subdivide their huge namespace into those you trust and those you do not.

2.2.3 Access Restrictions

For those Office of Science labs that perform even a small amount of classified work, even if not on the computing and data systems involved in open science environments, DOE is requiring that all foreign nationals and non-employees, including remote "cyber only" users (no on-site presence) who require access to lab cyber resources must have the appropriate Non-Employee Processing (NEP) system authorization/approval. On-site presence for this function is defined as an active, non-visitor badge for the lab. At ORNL, this has caused problems because visitors that

touch an experiment must get an ORNL badge when they are on site, and then when they leave, their status is not non-employee. Such inconsistencies will always arise when a population is divided into parts with no gray boundaries. How should people from other labs be treated? What is the exact definition of "cyber resources?"

This regulation implies that DOE needs to "trust" (at some level) all of its computer users. Is this necessary? It is clear that DOE is trying to protect its resources. However, there are examples of cyber access that require no trust and that are quite safe. For example, the touch-screen airport information systems can be used by criminals and terrorists without any extra risk. We all use the computerized phone access systems to direct our call to the appropriate party, and the only extra risk is to our patience.

Such "safe" systems all have one thing in common: the software and input mechanisms are severely constrained. In situations where these control mechanisms apply, it should be possible to allow access without extending trust.

2.2.4 Can You Control Access If There Is Any Access?

It is very hard to stop remote access if you allow any. Groove [14] is a collaborative environment that encrypts all traffic and files and claims to penetrate firewalls by using http if necessary. It successfully penetrated the PNNL firewall without any difficulties. Virtual private networks (VPNs – not the official lab ones) use network address translation (NAT) transparency mode (IP Security Protocol [Ipsec] tunnels that look like http traffic) to trick firewalls into believing that the traffic is really http. A VPN at ORNL penetrated the Thomas Jefferson National Accelerator Facility (Jlab) firewall as if it were not there. Finally, remote computers (the Grid, CORBA services) *must* penetrate firewalls if there is to be any distributed computing for DOE labs.

One great competitive edge of the USA in the cyber arena has been cheap, available remote access to remote computers. We must to preserve this in order to not hamper innovative approaches to science.

2.2.5 Encryption Changes Things

Soon all network traffic will be encrypted, solving the user's "network security problem." (However, the problem of securing the network infrastructure remains.) A proposed DOE policy is that DOE should be able to decrypt everything into and out of DOE Labs. This might be reasonable, except that because encryption is used in many net applications (PGP, S/MIME, Groove, SSL, PCAnywhere), it is impossible to enforce. The more worrisome problem is how can one detect attack or theft in an encrypted stream? Groove, for example, replicates files across a collaboratory, encrypting the traffic, the files on the hard disk, and even the file names. Therefore, the security vulnerabilities and protections occur at the clients, and not in the network infrastructure. Today, even viruses (e.g., Hybris) are being encrypted, making them harder to detect.

2.2.6 Virtual Private Networks Are a Partial Solution

The ORNL VPN implementation (Compatible/Cisco) has free clients for most platforms and can use Radius or PKI authentication. This VPN can create static tunnel addresses so lab users can access a Dynamic Host Configuration Protocol (DHCP) home machine from work. In addition, it

can use groups that are allowed to access only certain resources at the lab (used for subcontractors). But the use of VPNs opens a large security hole: there is no guarantee of the security of the remote machine. Microsoft was hacked through their VPN from a compromised home PC.

If the computer is at the lab, then in principle the machine can be placed under central control and they can be subjected to relentless scans, software control, etc., to reduce vulnerabilities But the remote user's PC is an unknown quantity and must be assumed to be "hostile." It may have viruses, worms, Trojan horses, keyboard sniffers, and remote users (Gnutella, the SETI screen saver, networked family members), all of which can attack the lab through the VPN tunnel. Can we allow access from such a platform without compromising DOE or its resources? The answer is that we *must* find out how to do this or else access will be severely restricted, thereby reducing the effective application of computing to science.

What sort of requirements should we place on such secure remote access? Some of the issues that must be faced are:

- Strong authentication. Is it really who you think it is?
- Use of encryption or secure hashes to prevent man-in-the-middle attacks.
- Ability to control which resources are accessed.
- Knowledge of what software is being run remotely.

One example of the weakness of current security measures is the use of Secure Shell (SSH) for remote access. SSH has strong authentication (can be certificate based) and encryption. However, SSH does not have the ability to control what resources are accessed, or the knowledge that a non-modified SSH (unhacked) is being run on the (remote) client. Because SSH has full access to the host machine, a Trojan version of SSH (which subverts strong authentication by stealing the user's identity) could be launching attacks in the background (an increasingly common form of attack in university environments).

One solution to this problem is to use custom client/server software. One way of doing this is to download a signed Java applet that must be used for access by, e.g., providing and/or authenticating the SSH client. In addition, there needs to be some way of knowing that the program being used is the one that you downloaded.

2.2.7 Scalability Issues

The current security infrastructure does not scale well to

- greatly increased bandwidths
- massively parallel distributed computing
- encrypted traffic
- distributed attacks
- computers that change their operating systems
- embedded operating systems that have no ability to control their security (e.g., unable to put DOE security warning banner in Axis camera servers).

Biological systems seem to scale well. Stephanie Forrest [15] has created computer analogues of most of the body's biological defences; they use more empirical approaches for anomaly detection and intruder eradication. However, it is more difficult to detect what is abnormal in a

research-oriented computer systems that tend to change their function and configuration fairly often.

Maintenance of security defences is resource intensive. Applying patches to reduce system vulnerabilities can be a full-time occupation if you are responsible for many systems. For Windows, Microsoft's update service (http://windowsupdate.microsoft.com) is a big step forward, although the update patches often are up to a month late. Application patching is harder because users are unaware of the vulnerabilities.

2.3 Technical Issues Raised by Security Policies

Sandy Merola, Information Technologies and Services Division, Lawrence Berkeley National Laboratory

The Department of Energy's mission encompasses fundamental science, energy research, environmental restoration and waste management, and national security. These mission areas motivate programs that are conducted by a widely distributed laboratory and university research community. Thus, there exists a widespread need for information collaboration (including high performance computing and networking, and other information technologies) and an underlying safeguards and security strategy. In particular, DOE and its laboratory and university community must establish cyber-security policies that enable and protect the success of the DOE mission.

The DOE has generated a substantial number of cyber-security directives (codified policies) during the last 18 months. A list of such directives can be found at: http://www.directives.doe.gov. Some of these directives have been well formulated to facilitate local cyber-security programs consistent with DOE and local institutional missions. Other directives have mandated specific cyber-security requirements such as the exact wording of "warning banners" or the number of digits that should be in a password. These types of directives tend to bring an emphasis on compliance-based security performance rather than performance based on effectiveness of response to the cyber-security threats of open science environments.

2.3.1 Cyber-Security Policy

Cyber-security policy, whether at the DOE community level or at the local institution, must be mission specific. Consider, for a moment, the difference between the missions of a defense institution and those of a university or open research laboratory. The safeguards and security policies that govern physical access and cyber security for those institutions must be substantially different. Moreover, the high-level DOE policies must motivate appropriate mission-related differences in local cyber-security policies. In every case, cyber-security policies should be intended to minimize security-related occurrences that detract from the mission.

It is the responsibility of the cyber-security community to provide adequate protection, timely detection, and appropriate reaction in support of protecting information technology resources. When a specific cyber-security weakness is exploited, policymakers frequently attempt to minimize that risk with a reactive policy. Thus, an acceptable level of cyber-security performance will reduce the numbers of such reactive, and too often poorly formulated policies.

More importantly, mission-appropriate policies must motivate the appropriate utilization of cyber-security technologies. The selection of technologies must follow the identification of risks, a cost/benefit approach to mitigate those risks to an appropriate level, and the formulation of supporting policies. It is only at this point that technologies can be selected to implement those policies.

The selection of policies and technologies frequently requires mutual iteration. For example, consider the case where there exists a policy mandating the characteristics of clear-text passwords. Suppose that the utilized information technologies have evolved to the capability of encrypted authentication. In this case, there should be a consideration of policy change that does not permit the usage of clear text passwords. As a second example, consider a requirement for a combination of high bandwidth network access and a very complicated set of access permissions that together exceed the performance of currently available firewalls. In such cases, either the policy must be changed so that a less complicated set of permissions can be implemented, or a different technology must be chosen (e.g., intrusion detection attached to blocking routers). In those cases where the combination of policies and performance requirements exceed available technology, then a motivation for cyber-security research and development results**.**

2.3.2 Protection, Detection, Response

The DOE complex includes a considerable breadth of unique and expensive facilities serving scientists around the world. Sophisticated demands on DOE facility utilization and network access exist concurrently, which increases cyber-security needs. This combination has continued to push the envelope of protection, detection, and response technologies. The advancement of these technologies is needed to protect and further the DOE mission. A few examples of mismatches between DOE cyber-security needs and cyber-security technologies are mentioned here.

In the protection area, the DOE community has been handicapped by the lack of availability of firewalls that can satisfy both high performance access and complicated access policies. Over the last couple of years, DOE scientists have sometimes been forced to give up needed functionality, such as video conferencing, because firewall technology has not kept pace with performance requirements needed by the scientific community. In other cases, some sites have chosen not to implement under-performing firewall technology rather than impose the associated handicaps on the scientific collaborations that the site must support.

In the detection area, the use of network-based intrusion detection has served to provide adequate levels of suspicious activity detection, but at an expensive labor cost and without sufficient ability for automated response. Current intrusion detection implementations provide an excessive number of false alarms. Thus, the judgment of a cyber-security expert is frequently needed to determine whether the suspicious activity is indeed an intrusion. R&D is needed to advance the ability of the intrusion detection software to more accurately identify potentially harmful activities, and enable the automatic reconfiguration of the blocking router to terminate access if that should be necessary. The automation challenge will get much worse as the bandwidths of network activity increase.

Also in the detection area, an Electronic Mail Analysis Capability has been implemented as a test pilot within the national security arena. The purpose of this pilot is to determine the

effectiveness of email monitoring as a mechanism to detect attempts to gather classified and other protected information through email communications. The anticipated success of this pilot will likely serve as an active deterrent toward such activities using email. Nevertheless, we can safely expect that those persons who are motivated to gather classified and other protected information will develop new techniques using different applications, prompting the need for the development of applicable deterrents.

In the response area, evolving policies and technologies are still in their infancy. Few policies exist that provide guidance to the process of evidence collection, and the information associated with both real-time and post-mortem investigations can be overwhelming and distributed. To make matters worse, the forensics gathering process can disturb the evidence itself and can, in real-time, affect the tactics of hackers-in-process. This area would benefit from research and development that facilitated increased rigor in the response process as well as provided tools that might help glean evidence from forensics data.

2.3.3 Cross-Cutting Issues Related to Future Cyber-Security Technologies

The disparate environment within the DOE would benefit from the development of security models that support the investigation of cyber-security issues associated with unique experimental facilities, high performance and data-intensive computing facilities, as well as high performance network interconnectivity. Models for open research environments as well as those environments requiring high security should be developed, as should models that allow either broad collaboration or isolation within individual institutions. These models are complicated by both the need for support of short-term collaborative environments as well as the interconnection of emerging mobile computing environments. Scaling issues associated with the Grid computing concept and its security requirements must be investigated to ensure that proposed approaches will be useful across the growing breadth of the scientific Internet community. The enclave model being heavily emphasized within the DOE community has implications that must be more clearly defined, and to do so will require the modeling of and experimentation with such environments. Even while the model of an enclave is not well defined, implementations of enclaves are occurring. Most importantly, cyber security must be implemented in a manner that allows the scientists to feel like they are operating in a local desktop-like environment. Models that serve as a focus to identify and conduct needed R&D for disparate environments of our community must be developed.

Our experience to date is that security is not designed into systems. Awareness of the importance of security considerations during systems design needs to be increased, and tools and techniques to simplify this process must be developed. It is not at all clear that applications designers, for desktop systems or distributed software, are motivated or have the techniques to develop secure components. Unfortunately, it is much harder to apply security around an application than inside or under it, so the application creator is the best one to apply the security methods. R&D must be conducted to allow investigation into the characteristics and design approaches that simplify securing software.

Additionally, a technique that would help motivate cyber-security-conscious design is the development of cyber-security evaluation criteria for software. This would require the development of a formal testing methodology. One beneficial result would be the minimization

of the proliferation of software with security exposures. This is a complicated issue as one begins to consider not just the evaluation of out-of-the-box systems, but of a fully functional environment. A *Consumer Reports* approach to the development of criteria as well as the distribution of the software cyber-security ratings would benefit from R&D.

Self-assessment and peer review of DOE and laboratory cyber-security implementations have already increased the effectiveness of our programs. Such assessments would benefit from agreed upon metrics of success. As performance metrics are developed, one can expect the current usage of compliance-based metrics would be decreased. Such performance metrics could be developed via an R&D focus area.

The human factors associated with cyber security give us both our greatest vulnerability and greatest potential improvements in this area, and will have high impact on our cyber-security environment. An understanding of the inconveniences that our users are willing to suffer to ensure security must be achieved. Cyber-security efforts to date are often thwarted by human actions, such as password handling or the improper configuration of firewalls. A best practices program would need to be underpinned by applicable R&D, and the implementation of such a human factors-based program would provide broad improvements to the security of our community.

2.3.4 DOE Motivation

An effective DOE cyber-security program requires the creation of policies that motivate and protect mission achievement. The selection of policies must be undertaken with a cognizance of the capability of existing technologies. Requirements for policies without the supporting technology will motivate the need for technological research and development.

DOE's unique facilities, and therefore unique requirements motivate such unique and self-serving cyber-security research and development. Further, as a major citizen of the Internet, DOE has a responsibility to contribute toward the advancement of technologies that can be shared by DOE with the entire Internet community. This contribution will benefit DOE by early deployment of such protection mechanisms within our own environment.

Part II: What Does the Future Hold?

3 The computing environment five years out

3.1 Computational and Data Grids

William Johnston, Lawrence Berkeley National Laboratory

"Grids" (see [16], [7], and [17]) are an approach for building dynamically constructed problem solving environments using geographically and organizationally dispersed high performance computing and data handling resources.

The overall motivation for the current large-scale (multi-institutional) Grid projects is to enable the resource interactions that facilitate large-scale science and engineering such as aerospace systems design, high energy physics data analysis, climatology, large-scale remote instrument operation, etc.

The vision, then, for computing, data, and instrument Grids is that they will provide significant new capabilities to scientists and engineers by facilitating routine construction of information based problem solving environments that are built on-demand from large pools of resources. That is, Grids will routinely – and easily, from the user's point of view – facilitate applications such as:

- coupled, multidisciplinary simulations too large for single computing systems (e.g., multi-component turbomachine simulation – see [18]
- management of very large parameter space studies where thousands of low fidelity simulations explore, e.g., the aerodynamics of the next generation space shuttle in its many operating regimes (from Mach 27 at entry into the atmosphere to landing)
- use of widely distributed, federated data archives (e.g., simultaneous access to metrological, topological, aircraft performance, and flight path scheduling databases supporting a National Air Transportation Simulation system)
- coupling large-scale computing and data systems to scientific and engineering instruments so that complex real-time data analysis results can be used by the experimentalist in ways that allow direct interaction with the experiment (e.g. Cosmology data analysis involving telescope and satellite interaction, and coupling to simulations)
- single computational problems too large for any single system (e.g. extremely high resolution rotocraft aerodynamic calculations)

Functionally, Grids are tools, middleware, and services for

- providing a uniform look and feel to a wide variety of distributed computing and data resources
- supporting construction, management, and use of widely distributed application systems

- facilitating human collaboration and remote access to, and operation of, scientific and engineering instrumentation systems
- managing and securing this computing and data infrastructure

This is accomplished through a set of uniform software services (the Common Grid Services – described in more detail below) that manage and provide access to heterogeneous resources. These services may be summarized as:

information services for resource discovery and management	resource specification and request
resource co-scheduling	data access
authentication and authorization	security services
auditing	monitoring
global event services	global queuing
data cataloguing	resource brokering
collaboration and remote instrument services	data location management
communication services	fault management

One of the key Grid services is security. The Globus implementation of Grids provides a common authentication model for all Grid applications. This model is typically based on X.509 identity certificates, but can also make use of Kerberos authentication.

The X.509 identity is the basis of two important mechanisms: single sign-on and a global identity for use by local resources. Single sign-on is accomplished through the use of proxy certificates. These certificates are derived from the user's authenticated identity, and are passed to remote systems to authorize service on behalf of the user. A common example is authorizing the submission of a job to a batch scheduling system on a supercomputer. The Grid identity mechanism uses a mapfile at each resource that maps the global identity to a local identity. This allows local resources to maintain local control, since the local identities are created and managed through the usual system administration mechanisms. The Globus job initiator is effectively an authenticated inetd daemon that first verifies the user's access rights on a system, and then creates the job/process requested by the user.

There are versions of SSH (secure shell/remote terminal) and ftp that use this single sign-on and authentication mechanism, and there is a basic secure messaging library that applications can use for secure interprocess communication. These mechanisms thus provide a powerful integrated set of security services for the Grid/Globus environment.

Figure 5 illustrates the high-level architecture of a Grid.

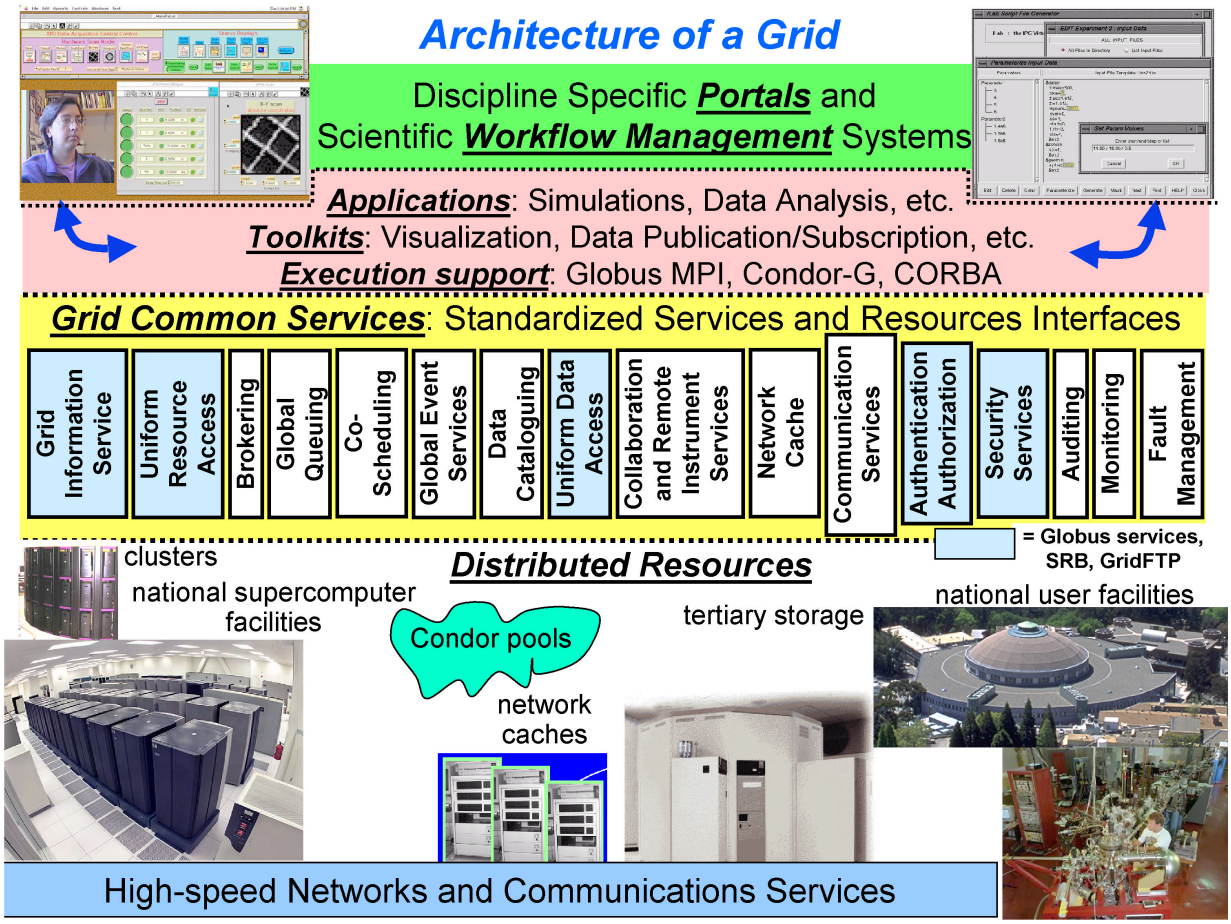


Figure 5. The Architecture of a Grid
Many different applications make use of a comon set of middleware services that, in turn, manage many different computing, instrument, and data resources.

3.2 Distributed Problem Solving Environment, Collaborative Workbenches/Frameworks and Portals

Dennis Gannon, Computer Science Department, Indiana University

Science "portals" are software environments for composing and controlling (remote) sub-components.

3.2.1 Example Scenarios

Chemical Engineering

A distributed group of chemical process engineers are working on a simulation of a new approach to copper deposition on silicon for semiconductors. One participant is a process engineer at a major company like Intel. Another participant is at a government lab who has a

simulation of the hydrodynamics of copper vapor in the deposition process. A third participant has a Monte Carlo simulation of molecular chemistry at the silicon surface. They want to couple the two simulations together and then add the process data from the Intel partner. How do they accomplish the task of linking the simulation codes? In particular, how do they do this in such a way that their collaboration is secure? How do the partners develop trust in the framework that supports their collaboration? How does the Intel corporation participant trust the remote computing environment enough to put Intel's proprietary process data into the remote simulations? How does the government lab partner trust the environment enough to allow the lab's proprietary simulation code to be used without compromising its internal operations? How do the partners easily set up a collective identity with the authorization to execute the assembled distributed application? The question of accountability for Cooperative Research and Development Agreement (CRADA) partners, and patentable data rests with both the collaboration (resource users) and the host laboratories/companies (resource owners).

X-Ray Crystallography

The standard process for scientist using a beamline at LBNL's Advanced Light Source or ANL's Advanced Photon Source is to get on plane and fly to the lab, work with the technicians to mount the sample, and return home to wait for the results. The xPort project[a] developed a set of distributed software components that work in conjunction with network-based video conference tools that allow the remote user the ability to FedEx the sample to LBNL or Argonne and then work interactively with the beamline technicians to conduct the experiment and analyze the data in real time. However, there are serious security issues here. One is physical safety: How do we allow remote users or automated processes access to the controls of a complex or dangerous instrument? For many remote applications the data sample may come from a hospital patient or from some other source that must be protected. When distributed applications operate on this data and Grid caching is used, how do we assure that the cached data are kept private? Are there temporary files generated by remote applications that cause information to be leaked?

3.2.2 Portals and Rapid Prototypes

The key to allowing users the ability to create collaborative, distributed Grid-based applications that solve problems like the ones above is

- make it very easy to assemble and connect the application components
- provide assurance that the infrastructure automatically supports their concerns for privacy and that the host Grid is assured of the authentication and safety of the collaboration
- make it very easy for them to encapsulate the results as metadata for specific executions including application parameters, scripts which launch and manage remote application components, execution event histories, and output file locations and characteristics.

The Science Portal project[b] is designed to provide the user interface to Grid applications that addresses the issues above. A science portal is based on a conventional Web server technology which has been supplemented by tools that allow the user to authenticate with the Grid services

[a] http://www.cs.indiana.edu/ngi

[b] http://sharan.ncsa.uiuc.edu/chemengathome/home.page.docs/NCSAPortalPlanning.FY01.htm

from an https transaction from a Web browser. This is accomplished by allowing the Web server to obtain a proxy certificate from a proxy server using the MyProxy protocol[a]. With the proxy cert, the server is now able to launch applications and interact with Grid services on behalf of the user. If the most common complex Grid interactions (such as launching multiple coupled applications or complex parameter sweeps) can be scripted, then the server can execute these operations on command from the user, who only has to fill out simple Web forms at the browser. The NCSA science portal contains a script engine that uses the Argonne Commodity Grid (COG) toolkit[b] to access Globus. It also has a metadata database to store application scripts and pages.

3.2.3 Software Component Architectures

A common solution to building distributed, multidisciplinary applications rapidly is to use a software component architecture that allows programming by composition.

In a component architecture, applications are built by linking together component sub-applications. A component is a software module that presents a set of well defined, data type-safe interfaces to the outside world. When a component runs, these interfaces are the only way to access the component's functionality. Some component interfaces are designed to provide services to other components, and some interfaces are designed to allow a component to use the services of others. In particular, a "uses" interface on a component running on one system on the Grid may be connected to a "provides" interface on a component on another system by using a secure, authenticated remote procedure call. This type of "encapsulation" has important security implications. Consequently, component-based software design is a significant and effective way to build distributed applications in which security concerns are addressed directly by the applications architecture.

In the commercial world, component architectures are becoming the standard for building enterprise distributed applications. Technologies like CORBA Components and Enterprise Java Beans are important new standards that have been designed with security in mind. These components may be "wrapped" legacy code or they may be new. Within the DOE, there is an existing effort called the Common Component Architecture to adapt these technologies to the high performance scientific application that are designed by laboratory scientists.

3.2.4 Conclusions and Further Observations

Grid collaboration will require the ability to rapidly assemble prototype distributed applications that can be shared within a dynamic security domain that protects both the users' intellectual property and the resource providers' resources.

Grid applications will need to leverage advances in Web and Internet technologies such as portals and software component engineering in which security is an attribute of the design methodology.

---

[a] The MyProxy server (http://dast.nlanr.net/Projects/MyProxy) manages the user X.509 credentials on a secure server. The user logs into that server and requests a proxy certificate that can then be used by Grid processes running on other systems.

[b] http://www.globus.org/cog

As we evolve from our current client-server models, future applications that involve distributed collaborative systems will resemble peer-to-peer systems, that is, each participant is both a client and a server. This will present an additional challenge to research for security for DOE, which has security requirements that greatly exceed current peer-to-peer frameworks.

4 Threats, Protection, and Security

4.1 The Future Threat Environment

Paul Krystosek, CIAC (Computer Incident Advisory Center), Lawrence Livermore National Laboratory

To understand future threats, we must study past and current threats with the intent of learning from them. From that study we can then develop techniques to lessen the effects of the threats.

4.1.1 DOE-CIAC Current Threats

Insider

Human beings tend to generalize problems and protect against the generalization. This works adequately for external threats: "Keep the evil hackers out of our network." It is difficult to generalize the insider threat and even more difficult to protect against it. Adapting existing protection techniques to the insider threat results in treating all insiders as if they were on the outside and, hence, not trusted. Our best course of action is to identify what must be protected and implement appropriate protections independent of the threat location.

Outsider Who Just Got In

Some sites depend totally on their perimeter defenses, enhancing the distinction between insider and outsider. It may be very difficult to get past such defenses, but once an outsider does, the rewards are significant. The former outsider now has insider status with all the rights and privileges so accorded. Users and system administrators in such an environment are not accustomed to looking for signs of insider misbehavior and so may not know to take action for some time.

Malicious Code

It is often useful to look to the past to predict the future. Malicious code (a term that encompasses viruses, worms, and Trojan horses) was a significant threat in the past and there is no reason to believe that the situation will improve. Protection against malicious code has traditionally been reactive: identify the signature, update the protection program. Writers of malicious code have consistently found new ways to defeat or bypass such protections. Without a significantly different technique for protecting against malicious code, we must assume it will continue to be a threat.

Unpatched Vulnerability

The reasons are not always clear – it might be complacency on the part of experienced system administrators, the ease with which a non-experienced user can install a system, or perhaps the

financial pressure to contain costs – but the fact remains, many systems are accessible via the Internet with significant vulnerabilities. Two specific categories of systems that are often vulnerable are the system taken "out of the box" and put into service with no regard for security, and devices such as scientific instruments that contain computer systems.

The out-of-the-box problem is one of convenience and, unfortunately, lack of responsibility on the part of vendors. Several brands of computer systems are still shipped with known vulnerabilities for which there are known fixes. The installation process has been made so convenient that users don't see the need for assistance, nor do they realize the insecurity of the system they just installed.

The other category is that of the instrument built around a computer. One problem is that any patches made to the computer portion of the system can render the whole instrument inoperable. The other is that the computer may be so embedded in the system that the owner does not have access to it for purposes of patching. Solutions to this problem include external protections or some form of isolation.

Newly Discovered Vulnerability

One problem we may never solve is that of the newly discovered vulnerability. There will always be a window of time during which any system with the vulnerability is open to attack. The vulnerability must be identified, the word spread, and a fix found and distributed.

Unprotected Passwords

Another instance of "You'd think they would have fixed this by now" is passwords. Even though several secure password protection techniques exist, many users' passwords are vulnerable.

Lack of Awareness

One of the biggest problems of all is the lack of awareness on the part of users. Excuses abound: "It is too much trouble," "I didn't know it was a problem," and so on.

4.1.2 Anticipated Threats

Mobile Code

Small, portable, connected devices often do not have the capacity to store all of the code needed to perform their functions. Mobile code is transferred to the device and executed on the user's behalf and then discarded when no longer needed. Similarly, Web browsers can load and execute code. This provides a new environment for the outsider to write malicious code and distribute it. Mobile code is probably not amenable to methods used for virus/malicious code checking, since there is a difference in intent. Mobile code is intended to be run on the device. Malicious code is often disguised as something else. The concept of not permitting external code to execute would help solve the problem, but at the cost of defeating the purpose of mobile code.

Mobile code can be digitally signed, but we must have the capability to determine that the signer is someone we trust. Recently there have been reports that techniques exist to steal signing keys, which would negate the trust relationship.

Mobile Workers

Most workers are not the threat, per se. We must accommodate working from home or on travel. This includes authentication, access, resources, and securing the mobile/remote computer.

Wireless

How does the use of wireless devices affect security? One way to think about it is that the "wire" is made public. There is no concept of physical security for most wireless communication. All protections must be in the data. Many systems have good protection available, but it is often defeated by lack of correct configuration. This is similar to the out-of-the-box problem discussed above – it works when first set up, so there is no further thought. Someone else can purchase the same equipment and potentially join the wireless network if within physical proximity. In any case, the signal can be captured and at least some rudimentary traffic analysis performed. The possibility of a denial-of-service attack by signal jamming should not be discounted.

How Does Moore's Law Affect Security?

Faster computers permit both sides (good guys and bad guys) to perform existing functions more quickly and contemplate functions not previously feasible. The bad guys can stage bigger denial-of-service (DoS) attacks, perform wider and smarter scans, and crack password files much more quickly. The good guys have the opportunity to improve intrusion detection systems (IDS), virus scans, and firewalls.

4.1.3 CIAC Input/Output

CIAC gathers a tremendous amount of data. What information compiled from that data would make sites' computer security better? Currently provided information includes:

- CIAC Bad List
- Alerts and bulletins
- Summary data (how many scans, compromises, etc.)

Tools are needed to analyze this data in order to draw conclusions and formulate responses.

4.1.4 Hackers Have the Advantage

The hacker community performs both as a team and individually. The rivalry and intense desire for publicity and/or acceptance motivates them to improve their skills and enlarge the body of knowledge. They have their own publications, Web sites, conferences, and communicate in real time via Internet relay chat (IRC). From the hacker's point of view, there are millions of potential victims and thousands of vulnerabilities to exploit. Until the social and legal systems change, there is little restraint from personal ethics or fear of punishment.

4.1.5 The Public's Disadvantage

The general Internet public is at a disadvantage in that hacked sites often do not want to talk about it, thereby depriving the rest of the benefit of experience. From the public point of view, one has to protect all those millions of systems against all those thousands of exploits. All of that costs money and takes time.

As much as we would like, we should not retaliate against the perpetrators. We are not on a war footing, and any response more aggressive than terminating connections set up by the attacker would likely put the defender in violation of whatever laws applied to the attacker.

4.1.6 Be More Like a Hacker?

If it works so well for the hackers, why not do it? Share information on intrusions, find a way to distinguish the routine from the really important, and treat both accordingly. Actively protect your turf and help others to do the same. This sounds easy, but can be very difficult to implement due to the scale of the problem: both the number of systems and instruments, and the rapid discovery of new vulnerabilities and forms of attacks.

4.1.7 Don't Be Surprised

If we communicate, there will be fewer surprises. Make time to patch systems. Install intrusion detection and watch for irregularities. Actively look for vulnerabilities in your own systems. Systems exist that automate the patching process; use them.

4.1.8 Ethics

It is not at all obvious how, but we must attempt to instill ethics in hackers. Their skills are more valuable put to work for the public good than for their own purposes. Education and public pressure work slowly but tend to be relatively permanent.

If possible we should find ways to ethically retaliate. But in the meantime, do not hit the wrong target; make it hurt; make hacking less fun, less gratifying, and less profitable.

4.1.9 Cyber Undercover

Know your enemy. Learn what is known about their motivations. The more we know about the opposition, the better we can protect ourselves. We must realize that there is a hierarchy of hackers including:

- script kiddy
- talented amateur
- professional
- foreign government.

Learn how they are different, how to identify them, and how to protect against each one. The problem with this is the system administration skill and time required for this detracts from productive work.

4.1.10 Research Project Ideas

"They got guns, we get guns." (loose quote from *West Side Story*)

Translation: Make hacker tools available and explain their use.

Hacker Tools

If you have used a hacker tool on yourself, you are more likely to recognize its use against you. They are cheap, and some of them are quite good.

How to make computer security de rigueur for users? What can we learn from other efforts?

For example, seat belts were introduced a long, long time ago. For some time, the average driver did not understand their significance. As the message was spread wider and more insistently, they started to catch on. Today their use is a matter of law, though some dispute them with various weak arguments.

We learn slowly, but it can be done with a persistent, consistent message. If we assume that computer protection will be similar, then we must start now if it is ever to happen.

Part III: What Needs to Be Done? R&D Topics

5 Scientific Collaboration / Collaboratory Issues

Steve Tuecke, Mathematics and Computer Science Division, Argonne National Laboratory
Keith R. Jackson, NERSC Division, Lawrence Berkeley National Laboratory
William Johnston, NERSC Division, Lawrence Berkeley National Laboratory

5.1 Security Considerations

When building security solutions for collaborations, one must consider:

- *Level of protection:* Various collaborations may require various levels of protection, both for information and for resources. For example, many DOE collaborations include large, expensive resources which must be protected from misuse and damage, as well as cheap resources such as the individual PCs of participants. These differences must be taken into account when developing security solutions. Further, individual researchers may work in multiple collaborations which require different information protection levels; in fact, some national laboratory researchers even work in a mix of classified and unclassified collaborations.
- *Scaling:* Collaborations span from small (two parties) to large (thousands of participants and resources), and often include international membership.
- *Dynamic vs. static:* Collaborations range from ad hoc (e.g., phone calls), to long lasting and computer mediated.
- *Ease of use:* If security is hard to use, then it will typically either prevent or hinder the collaboration, or the users will misuse/circumvent the security system (either purposely or accidentally), thus causing security failures.
- *Cost of deployment:* Different collaborations may have different sensitivities to the costs of deployment of a security solution. Such costs include hardware and software costs, administration costs, and training costs. If it is a small, ad hoc collaboration, the cost should be minimal. Large, sensitive collaborations may tolerate much higher costs.

5.2 Accountability

Central to the problem of security for collaborations is *accountability*. For example, various collaborations have requirements for accountability of use of information (e.g., data that is proprietary, patentable, covered by a CRADA, etc.) and of resources (e.g., computer cycles, storage space, device usage, etc.).

Further, individual members of a collaboration may have different accountability concerns. Commercial and university participants may have different concerns regarding information usage, and resource users may have different concerns than resource owners.

In general, accountability takes several forms:

- *Authorization*: Determining who is allowed to use collaboratory system, and then enforcing that decision.
- *Accounting*: Determining whom to charge for cycles, storage, data, etc. that are consumed as part of the collaboration.
- *Auditing*: Determining, usually after the fact, who did what. This ranges from legitimate resource usage (usually for accounting or allocation management) to determining what was done during a security compromise or failure to assist in recovery or for forensics.

Each of these types of accountability require *authentication*, which is the establishment of the identity of the party using the system. Research and development is required in each of these four A's of security, in order for them to be effectively deployed in collaborations of interest to DOE.

5.2.1 Authentication Research

Authentication within a collaboration is significantly complicated by the fact that many collaborations span multiple security domains, each of which typically has differing security policies and practices. This raises numerous questions which require further R&D, including:

- Is ubiquitous authentication doable?
- Does it require DOE-wide or worldwide unique identifiers, or management of many identifiers? Are these identifiers issued by multiple authentication authorities?
- How can the various authentication systems already deployed be integrated into a collaboratory's authentication system? These systems range from minimal to very high levels of security. Some are standard products, while others are homegrown. Some are based on secret key technologies (e.g., Kerberos), while others are based on public key technologies.
- What is a credential? For example, even within the public-key world, there are numerous approaches, including SPKI, X.509, and PGP, each of which has a different notion of "identity."
- What are the trust relationships upon which authentication is built? Is it between users and individual resources, or between centralized authentication authorities? (Again, SPKI, X.509, and PGP each have a different notion of trust.)
- What is the role of multi-factor authentication? Can the physical security measures employed by some DOE facilities be leveraged for cyber security?
- What role is there for anonymity and privacy within collaborations, and how does this affect authentication?

5.2.2 Authorization Research

A rich and active area of security research is in the expression and checking of authorization policies. For example:

- How to define policies regarding which users and resources can participate in a collaboration?
- How do sites define how they are willing to participate in a collaboration?

5.2.3 Accounting and Auditing Research

Currently accounting and auditing tend to be handled resource by resource, or organization by organization, and often involve homegrown tools. But when a collaboration spans many organizations, there is a need for common underlying mechanisms for accounting and auditing of everything in the collaboration for which we want accountability. This leads to the need for research and development in distributed accounting and auditing, including:

- common accounting records, which allow for the expression of use of the many different types of resources that may comprise a collaboration
- protocols for distributing accounting information to those parties who require the information, and who are authorized to see the information
- protection of accounting information
- analysis of accounting and auditing information.

5.3 Use of Untrusted Resources

A common problem encountered in collaborations is how to include an untrusted or compromised resource into a collaboration, without unduly compromising the rest of the collaboration. For example, from the perspective of the administrator of an expensive resource such as a supercomputer, the laptop computer of a remote user that is used to gain access to the supercomputer is likely an untrusted resource. While that administrator may be able to exercise great control on the supercomputer and have great confidence that it has not been compromised, no such confidence can be maintained over the user's laptop. Perhaps it has been compromised by a virus, or a key stroke recorder, etc. Such laptops and home computers have often, in fact, proven to be a backdoor onto laboratory networks. Yet administrators must allow access to the supercomputer by the user's laptop in order to enable that user to perform their required work

This tension between keeping resources secure, while allowing access to such resources from untrusted resources, is a rich source of security research and development topics. For example:

- *Validating untrusted environments*: Prior to granting access, one may wish to validate that the machine from which the access is coming is properly maintained (e.g., OS patches, running a personal firewall, who has root on the machine, etc). The specification and checking of such attributes is an open problem.
- *Sandboxing*: In order to minimize the detrimental effects of accidental or malicious misuse of a resource, a common technique known as sandboxing is used to limit the environment in which an operation runs. A common example of this is sandboxing of Java applets within a Web browser. However, there are many other instances where this general approach of sandboxing might be applicable, including: protecting a resource from a malicious program; protecting one process from another process within a shared environment; placing limits on things such as resource usage; restricting an operation to gaining access to only a portion of a resource, enclave; etc.

- *Restricted delegation*: In a distributed system, it is common for a user to delegate to other parties (i.e., other users, or processes/agents running on that user's behalf) the right for those other parties to act on behalf of the user. Yet, when performing such a delegation of one's access rights, one may wish to limit exactly what rights are delegated, so that they cannot be intentionally or accidentally misused.

5.4 Perimeter Protection

A common approach to protecting resources is through perimeter protection, such as through the use of firewalls and proxies, and via intrusion detection and reaction. The appropriate approach to perimeter protection is a question of many tradeoffs. For example, while the simplest approach to securing a resource may simply be to completely wall it off from the outside world except for a very small number of services such as Web and ssh, this approach is contradictory to the needs of advanced collaboratories, which increasingly require rich interaction between participants via TCP (Transmission Control Protocol), UDP (User Datagram Protocol), and IP (Internet Protocol) multicast. In addition, intrusion detection devices often have severe performance limitations, which hinder collaboratories that require high performance networking. Further research and development on adaptive, smart perimeter protection is needed to better understand the tradeoffs, and to provide alternative approaches that can accommodate the needs of collaboratories.

5.4.1 Perimeter Protection Research

One increasingly common need of collaboratories is to move data at high speeds between participants across organizational boundaries. Many communities need to move huge experiment and simulation data sets amongst community participants, while others need to move high-speed multi-media data streams. Common firewall proxies and intrusion detection systems can severely limit network bandwidth, often by as much as an order of magnitude. Better approaches are needed to allow for high-quality perimeter protection, but without crippling network performance.

Many collaboratories are also moving toward more complex interactions between participants, for example with multi-media data. Such traffic often requires UDP or IP multicast, both of which are commonly blocked by current firewall implementations. Research and development is needed on better ways to monitor all such flows while protecting the perimeter appropriately.

Real-time intrusion detection is an increasingly common method of perimeter protection. However, current systems suffer from high false positive rates, and also require that human administrators react to intrusion alerts. Research and development is needed on better analysis and identification of intrusions, as well as on automatic reaction to intrusions. Further, as collaborations become larger and more widespread, more work is needed in distributed intrusion detection approaches that take into account that what is "intruded" is now a semi-autonomous collection of facilities with poorly defined boundaries.

5.5 Scaling Trust Environments

DOE people and resources are increasingly involved in large collaborations, which may include thousands of people and resources. While various organizations have developed good

approaches to managing large user and resource populations within a single organization, better approaches are needed to manage such populations when they are spread across multiple security domains. Such approaches need to consider not only normal addition and removal of participants, but also timely and automatic recovery from compromises of identity.

5.6 Ease of Use

A hard-to-use security system tends to be an insecure security system. Users and administrators will either tend to make mistakes that compromise security, or will work around the security system, thus weakening it. The answer to this is not simply education, as scientists should not want to be security experts. Rather, strong security also needs to be made easy to use, so that it will be used correctly and consistently, ideally fading into the background of day-to-day work. In addition, security systems that are hard to install, administer, and use are not likely to be adopted rapidly.

Further research and development is needed to allow joining a collaboration to be easy – perhaps as simple as making a phone call. Questions to consider include: What mechanisms, if widely deployed, would allow for DOE-wide Grid services to be quickly marshaled to enable dynamic collaborations? What services or policies would make collaboration really easy? What applications must be integrated with the services?

5.7 Inter-Process Communication Research

Collaboratories employ a wide variety of communication, including parallel programming libraries such as MPI and PVM, multi-media flows, group communication (reliable multicast), and high performance data transfer. With each of these approaches, various tradeoffs can be made between level of protection and performance. Further research is needed on:

- Ways of specifying how and when to employ such tradeoffs. For example, knowledge of security domain boundaries may be exploited to allow for reduced security and increased performance between participants inside the same boundaries.
- Ways of protecting high performance flows, such as specialized encryption algorithms that exploit knowledge about the type of data or the type of communication used.
- Ways of protecting non-TCP flows, such as unreliable, out-of-order, and/or multicast flows.
- Group security protocols, which maintain protection despite people entering or exiting the group, network failures that partition the group, etc.

5.8 Grid Information Services: Naming, Discovery, and Cataloguing

The Grid will be a global infrastructure, and it will depend heavily on the ability to locate information about computing, data, and human resources for particular purposes, and within particular contexts.

Most Grids will serve virtual organizations whose members are affiliated by a common administrative parent (e.g., the DOE Science Grid and NASA's Information Power Grid); a common long-lived project (e.g., high energy physics experiments); a common funding source; etc.

A “Grid” might be defined by a common infrastructure, and/or by virtual organization criteria.

The Grid Information Service (GIS) is a key Grid middleware service that manages and provides state information about Grid resources:

- computing and storage system architectural characteristics
- computing and storage system operating state, including, e.g., user authorizations/allocations
- referrals to services associated with the resources – process state, network connectivity, etc.

Most of this information may be considered sensitive by system admins, and access control mechanisms must be provided.

The question of using a hierarchical structure versus, e.g., very fast searches on flat attribute spaces (like Web search engines), is an open issue, and these approaches may have very different security issues. However, here we are addressing the issues of a hierarchical directory structure.

5.8.1 GIS User Requirements

Searching

The basic sort of question that a GIS must be able to answer is, for all resources in a virtual organization, provide a list of those with specific characteristics.

For example:

> “Within the scope of the Atlas collaboration, return a list of all Sun systems with at least two CPUs and 1 gigabyte of memory, that are running Solaris 2.6 or Solaris 2.7, and where I have an allocation.”

Answering this question involves examining both architecture characteristics and state information in order to produce a list of candidates.

Virtual Organizations

Virtual organizations (VOs) enable disparate groups of organizations and/or individuals to share resources in a controlled fashion, so that members may collaborate to achieve a shared goal. It should be possible to provide “root nodes” for virtual organizations. These root nodes would sit at the top of a hierarchy of VO resources and serve as starting places for searches. Like other named objects in the Grid, these VO nodes might have characteristics specified by attributes and values. In particular, the VO node probably needs a name reflecting the organization name; however, some names (e.g., for resources) may be inherited from their Internet DNS (Domain Name System) names.

Information and Data Objects

A variety of other information will require cataloguing and global access, and the GIS should accommodate this in order to minimize the number of long-lived servers that have to be managed:

- dataset metadata
- dataset replica information
- database registries

- Grid system and state monitoring objects
- Grid entity certification/registration authorities (e.g., X.509 Certificate Authorities)
- Grid Information Services object schema.

Therefore it should be possible to create arbitrary nodes to represent other types of information, such as information object hierarchies.

All of this sort of information has to be consistently named in a global context, will have to be locatable, and in some cases will have an inherently hierarchical structure.

Requirements for these catalogues include:
- providing unique and consistent object naming
- access control
- searching, discovery, and publish/subscribe.

5.8.2 Operational Requirements

Performance and Reliability

The GIS plays such a key role in Grids that any problems with the GIS are likely to produce widespread disruption of service. Therefore:

- Local sites should not be dependent on remote servers to locate and search local resources.
- It should be possible to restrict searches to local resources of a single, local, administrative domain.
- Site administrative domains may wish to restrict access to local information, and therefore will want control over a local, or set of local, information servers.
- Queries, especially local queries, should be satisfied in times that are comparable to other queries like uncached DNS data, e.g., seconds or fractions of seconds.

These requirements imply the need for servers intermediate between local resources and the virtual organization root that are under local control for security, performance management, and reliability management.

(Note that in the Globus terminology, these intermediate directory servers are called GRISs, or Grid Resource Information Services.)

Multiple Membership

Many objects/resources will have membership in multiple virtual organizations. This information, like other resource attributes, will likely be maintained at the resources in order to minimize management tasks at the upper level nodes.

It must be possible for a resource to register with multiple virtual organizations because this is a common circumstance in scientific collaboration.

The requirements raised so far for the GIS are illustrated in Figure 6.

Security and Access Control: Control over Information Propagation

At each level of information management (four have emerged so far), there are various reasons why both import and export controls will have to be established. (See Figure 7.)

At the object / resource level, the local administrators must have control over what information is exported.

At the object / resource level, there must be access control mechanisms to restrict the types of queries, or the detail that queries return.

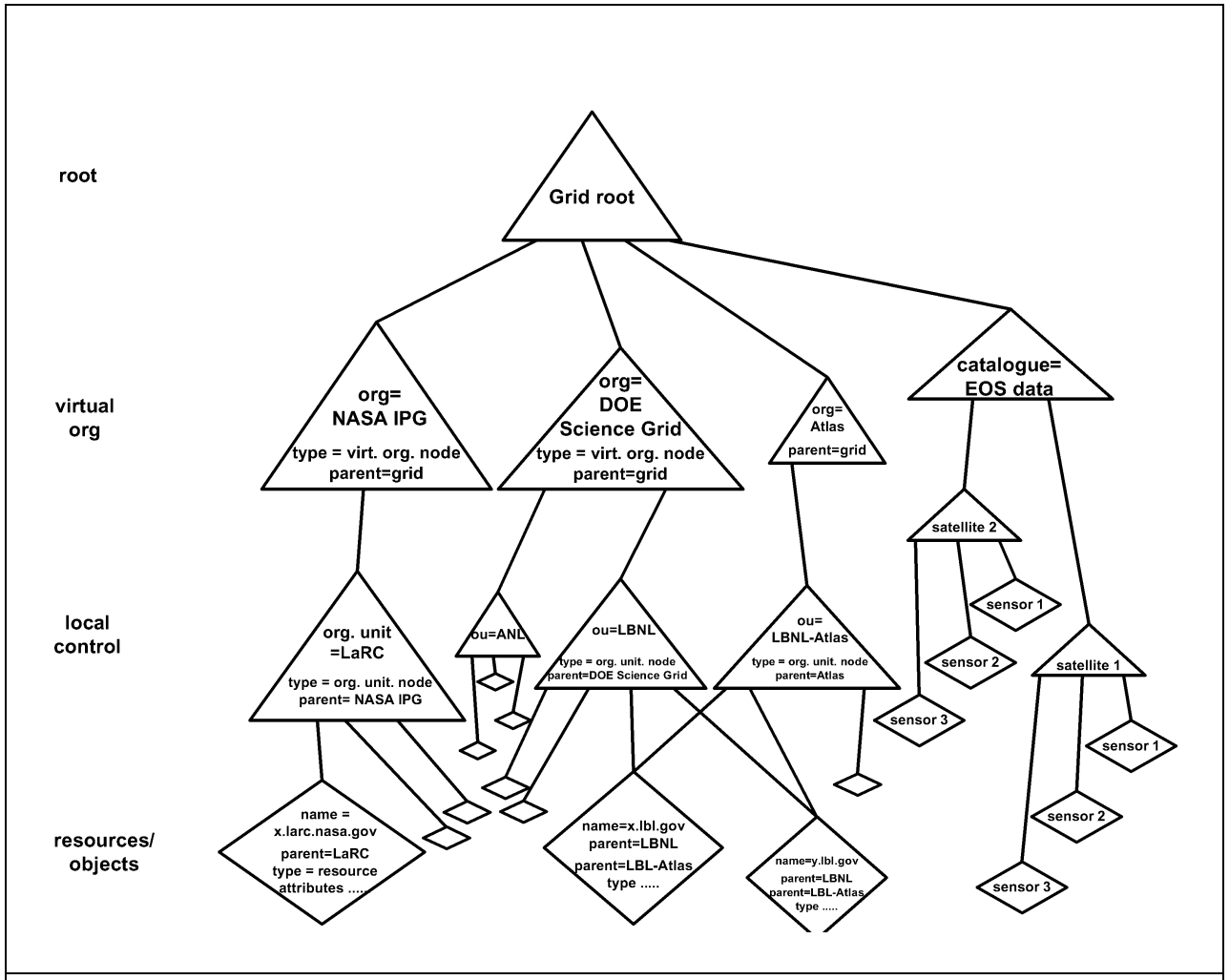


Figure 6. Characteristics of the Grid Information Service – 1

- Virtual organizations (VOs) may need to register with a Grid "root" so that they can easily locate resources within the organization.
- Local / intermediate nodes (e.g., GIIS [Grid Index Information Service]) between the VO root and the resource nodes may be needed for performance, reliability, and security.
- Membership in multiple VOs will require support for multiple parents.

The nodes at the level of "local control" are meant to model a common system administration domain, and must support a common security policy, including who is allowed to register

(import control) and what information is passed outside of the security domain (export control). It should, for example, be possible to implement policies such as making anonymous the information that is passed to the next level up (either for registration or as search results).

Such anonymous information should allow broad searches at the upper levels, but limit specific searches to the lower levels, where searches can be authorized based on the relationship of the searcher to the resource.

The same sorts of capabilities as exist at the local control level must be available at the virtual organization level in order to maintain control over the characteristics of the virtual organization.

At the root, again it must be possible to apply policy to registration (e.g., to prevent nodes below the VO level from registering at the root).

The ability to do automatic node replication for reliability must exist at all levels.

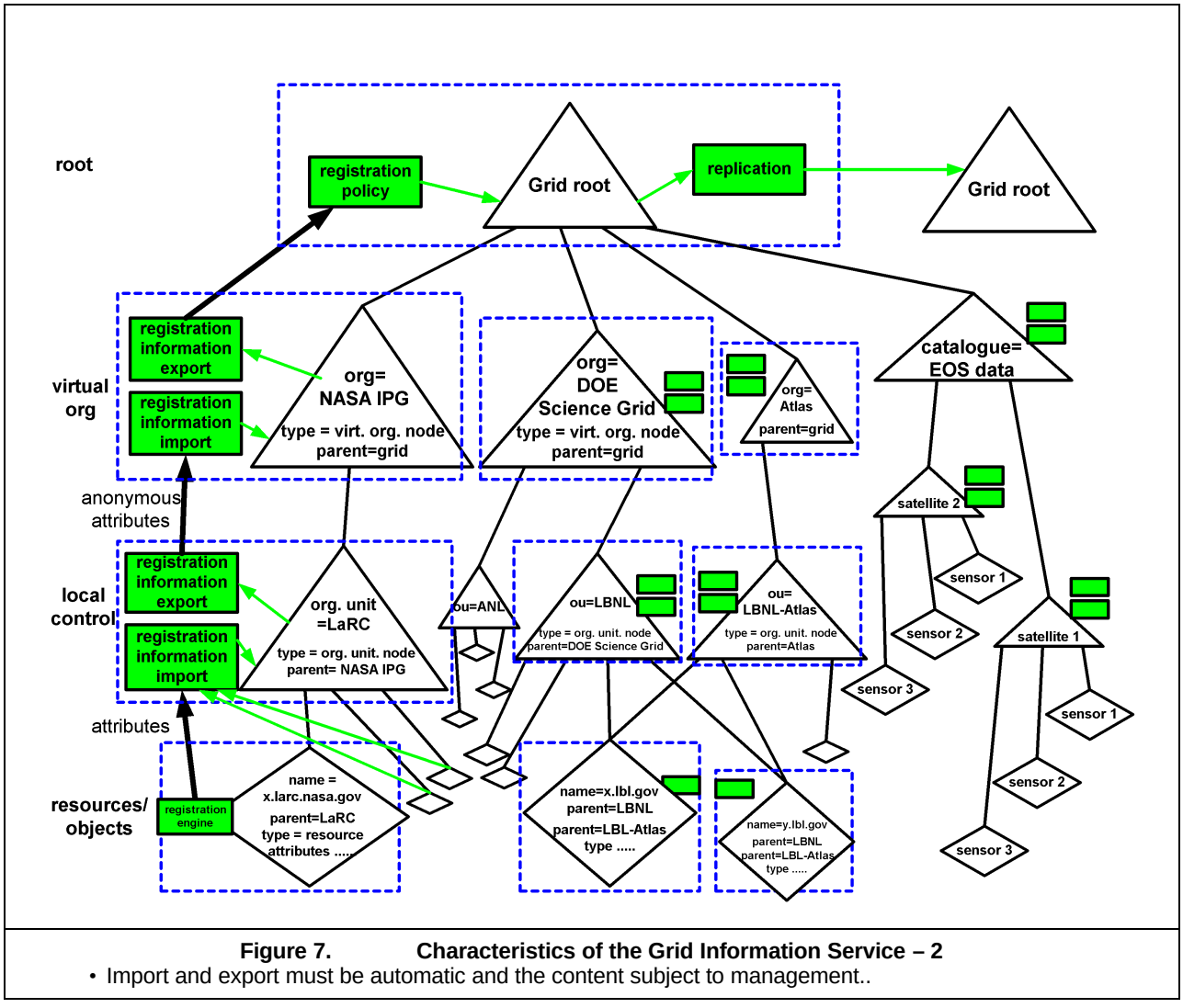


Figure 7. Characteristics of the Grid Information Service – 2

- Import and export must be automatic and the content subject to management..

These are all security research and development issues associated with the Grid Information Service.

5.9 Ratings, Metrics, and Analytical Models

A question commonly asked before using a resource is, how well is this resource protected? The answer to this question is usually qualitative. Standard metrics are needed which can be used to quantitatively rate and evaluate resource attributes such as quality of protection, scalability, policies, intrusion detection, etc. Such rating systems could then be automatically evaluated, for example, when a user is considering use of a particular resource.

In addition, research is needed into analytical models of security. For example:

- How can one model the effectiveness of an intrusion detection system?
- Can distributed attacks be modeled?
- Can a model be developed which gives a level of certainty that a particular observed behavior is an intrusion or a denial-of-service attempt? Such a model might be useful in improving the false positive rates of intrusion detection systems.

6 Collaboration Domains and Enclaves

James Rome and Walter Dykas, Oak Ridge National Laboratory
Jim Rothfuss, Lawrence Berkeley National Laboratory
J. D. Fluckiger, Pacific Northwest National Laboratory

Given the desirability of cross-organizational collaboration, it is important to have an objective and standardized method for evaluating the security policies of other organizations in order to establish trust relations between different administrative domains.

An enclave is a term that is being loosely used in the DOE research community to refer to a large number of complex resources that share a common security policy. One of the purposes of an enclave is to allow mutual trust among all the platforms within it. For cross-organizational collaboration, it might be possible to establish a cross-organizational enclave, or "collaboration domain," or to establish two enclaves that have the same security policies and thus can establish a clearly defined trust relationship between them. In order to talk about membership in an enclave for a particular machine, or trust between two enclaves, we need to have a way to characterize their security policies. Such policies have many dimensions, for example: methods of strength of authentication and authorization, file servers and the type of access control they enforce, OS version and patch level, network services that are available, security practices such as physical access to resources and enforcement of password rules and platform rules.

6.1 Collaboration Domains

Collaboration domains (CDs) provide an environment for the safe, protected exchange of ideas. Enclaves are a lab-based way of implementing or defining a set of security policies (although they may span several labs). Thus, a CD is generally composed of one or more enclaves. The CD may also impose additional security constraints as needed.

CDs are the mechanism for people to easily start, use, and maintain the security infrastructure for a collaboratory. As a goal, creating a CD should be as easy as making a phone call. But in fact, creating a security domain that does not necessarily coincide with the network topology of a lab, and that might span laboratories, is a challenging task.

The purpose of a CD is to host a scientific collaboration, so its security requirements will usually comply with but differ from those of any of the hosting institutions. For example, a collaboratory may contain users from different companies who cooperate with each other and use the same infrastructure, but who wish to maintain some proprietary information. Special security mechanisms must be in place to provide this additional protection. The collaboratory may require backup schedules to protect against data loss due to an external attack or user error, while such a requirement does not exist in any one enclave.

These special security features of the CD need to be protected by the enclaves comprising the CD, perhaps taking advantage of the hosting institution's security mechanisms. One challenge for the CD is that in general, it is a distributed collection of resources that is not accessed through a single point, and thus protection mechanisms might have to be host based. In addition, the CD may have to penetrate the outer perimeters of several different labs and be able to get through their perimeter defenses in a safe, allowed manner.

One instantiated example of a CD is the Groove collaboratory environment [11]. Some aspects of Groove are more successful than others. Groove successfully links its diverse members across domains and uses strong encryption to define and protect its boundaries. Even the files on the hosts are encrypted so that different Groove spaces cannot share information unless the user moves it explicitly. Groove claims to be able to penetrate firewalls by using NAT (Network Address Translation) transparency mode and proxy servers if necessary. However, Groove fails to enforce any policies about who can join the collaboratory space – any member can invite someone without the other members' approvals. Groove also silently updates all files to all members of the space, and because the files are encrypted, they evade the virus scanners of the host machines.

CDs are defined by the collaboratory members, while enclaves are defined by the labs. The goal (and challenge) is to obtain a set of distributed enclaves that allow the CD to function and to enforce its security policies. Each new CD should not have to start at this from scratch, because the time and effort required to comply with the policies and restrictions of multiple sites can be enormous. Therefore, one purpose of enclaves is to have a predefined set of diverse security policies at each lab that can be combined and utilized to allow a new collaboratory to be created with minimum hassle.

6.2 Enclaves

The purpose of this section is to define "enclaves" and to describe how they relate to one another as well as their relationship to collaborations.

The definition we will use is "a defined set of computing resources, the protection of which is the same and the security of which is managed as a single entity." Enclaves may be a single local-area network (LAN, a group of systems and the connecting communications links), a group of LANs, a wide-area network (WAN), etc., and may be located at a single site or spread over a

number of sites remote from each other. Examples of simple enclaves are shown in Figures A, B, and C.

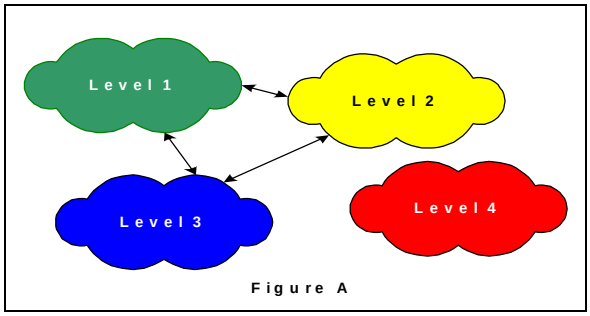


Figure A

Figure A shows several enclaves, each of which has a different level of protection, which are connected together with the exception of the enclave shown with security level 4. The unconnected enclave in our example might be a standalone, classified enclave. In each case, the enclave with the higher level of protection controls the connection between enclaves.

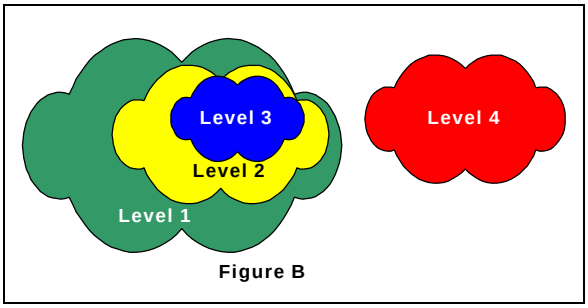


Figure B

Figure B shows nested enclaves where the level of protection increases as you go deeper into the networks. An example of this might be a single network where the Level 1 enclave has an open connection to the Internet with only an intrusion detection system (IDS) to protect it; Level 2 has a firewall between it and Level 1; and Level 3 has an internal firewall between it and Level 2 with strong authentication for workstation access.

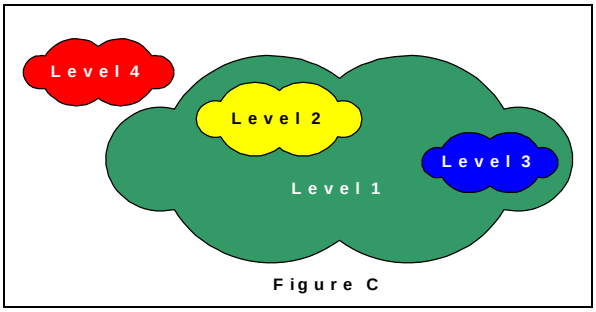


Figure C

Figure C shows a different form of nested enclaves where rather than going deeper and deeper into the network, the nested enclaves are physically at the same level in the infrastructure but have different levels of protection. An example of this might be a site with a single network where the Level 1 enclave has an open connection to the Internet with only an IDS to protect it; Level 2 has a firewall between it and Level 1 but is only used to protect against DOS (denial of service) attacks and network scans; and Level 3 has a firewall, possibly the same one, between it and Level 1 with a more stringent rule set for protecting the Level 3 enclave.

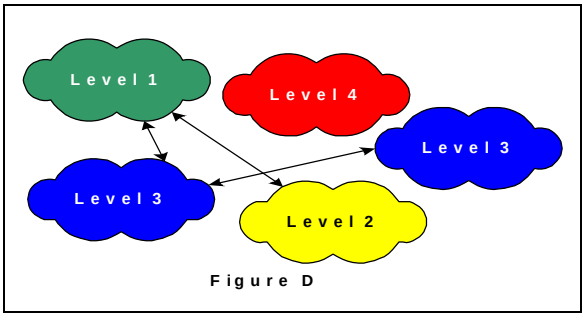


Figure D

Figure D shows that there may be numerous enclaves in a single network or at a single site and that some of them may have the same protection level. In those cases where the protection level is the same, the communications link between them may be open, implementing a trusted relationship between the enclaves.

What protection Level 1, 2, 3, or 4 means may vary from one company or organization to the next. Before it is decided how required communications links are to be put in place between these organizations, or within a

collaborative enclave (described below), the equivalency of the protection mechanisms may need to be negotiated so the parties know what level of trust to place on the link.

These are just some of the possible enclave configurations and relationships that might be developed to meet the needs of the site when it comes to protecting the computing resources. Any combination of these examples could be used if needed, as well as others not described here.

6.2.1 Collaboration Domains

Once we agree on a basic understanding of what enclaves are and how they might be related to one another, we can use the concept in setting up collaborations.

A collaboration domain (CD) may be put in place involving agreeing parties simply by linking enclaves of the same protection level together. This would require that the definition of the equivalent level of protection be the same at each of the collaboration sites. This is shown in Figure E by the configuration labeled "Collaborative Domain I," which includes the two level 3 enclaves within the striped orange area.

It should be noted that it is not likely that any site will voluntarily allow a piece of their network (enclave) to be operated and managed independently by another party as part of a collaborative domain.

Figure E also shows, within the gray dotted area labeled "Collaborative Domain II," a CD that involves only parts of enclaves at different collaborating sites. The plum area labeled "Collaboration Domain III" shows the same thing, only at a single site. In this case, besides having equivalent protection mechanisms at each of the sites, it would be necessary to accept the additional risk associated with having computing resources not part of the collaboration in the same enclaves with resources that are part of the collaboration. If this were not acceptable, an alternative would be to separate the resources out into their own enclaves.

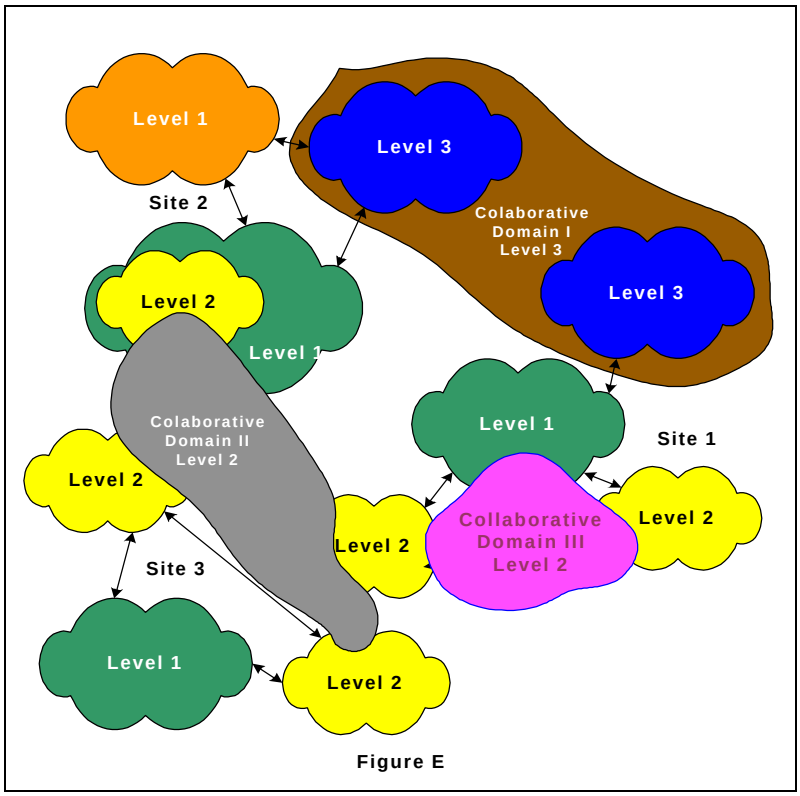


Figure E

When the levels of protection differ, as in CDs II and III, the level of protection that results from the collaboration will need to

be negotiated between all parties involved. Part of the difficulty here is that in general, for unclassified systems, it is difficult to define and maintain sharp boundaries for each level of protection. One way around this difficulty is to devise *protection level gateways* that allow safe connection between trust levels.

The number of sites that might participate in a collaboration domain built using enclaves as described above is not limited to any maximum number. The only requirements would be the need for equivalent protection mechanisms and the ability of the technology to include the number of collaborating sites who want to participate.

6.3 Research Topics

6.3.1 Motivation

An enclave should be easy to form, and the red tape necessary to form one should be codified and approved in advance among the DOE labs so that this can be accomplished. Because each lab is using a different approach to protect itself, this effort requires some sort of harmonization of the individual laboratory requirements.

6.3.2 Issues

A collaborative domain must be defined in order to instantiate it. The definition must encompass at least specification of the user community in the CD, the information and resources and their required protection levels, and the boundary of the CD. CDs might be composed of sub-enclaves (e.g., one at each lab) or might be a composite structure with its own boundaries. A CD can be static or dynamic. For example, a Grid resource might be temporarily used by the enclave and is for that time “inside” the CD. A scientist might be a member of several CDs. Is having the information of multiple CDs on one computer allowed? If so, how is information flow between the CDs controlled?

In addition to the membership and boundaries of the CD, its rules and regulations must be created. In the computer domain, these rules are known as the *policies* it must follow, which are distinct and different from the policies of the labs and DOE itself. If the CD is composed of many enclaves, the enclave policies must be consistent with those broader policies; but because the CD has special needs and requirements, it must have additional policies to define how these needs are to be met. A good example of an enclave policy is the requirements for a Kerberos KDC (key distribution center) put together by a team led by John Volmer of ANL.

Another aspect of the policy must be the determination of how new members and resources are brought into the CD and the enclave. How do the CD members know when the CD has changed, and how do you know if you trust their resources?

As part of the formation of a CD, its termination must also be defined. If part of a CD leaves, how its resources and information are dealt with should be predetermined, similar to a marriage prenuptial agreement.

Once the CD is defined along with its policies, how can we assure that they are always followed? What sorts of proof (e.g., auditing) are required to satisfy the members and the external stakeholders?

Scalability is another issue. CDs might range from just a few people to organizations across many labs. It needs to be easy to do tasks such as adding or removing members, and managing many enclaves.

6.3.3 Policy Needs

Agreeing on the cross-organizational procedures for trust is not an R&D topic, but it is absolutely essential to the success of any collaboratory, and hence to the formation of CDs. It is not attractive to scientists to have to spend time determining the requirements for each of many separate organizations, and it is even less enjoyable to try to fulfill them. In general, in an open (non-classified) research environment, if security is not made easy and couched in understandable and reasonable-sounding terms, it will be evaded, avoided, and not taken to heart. This is important, because security is often more about policy and procedures being followed than it is about the application of technology.

Therefore it behooves the security heads of the various labs to get together and to agree upon a common framework for intra-lab trust. Trust is a very difficult and often ill-defined subject. In the first place, not all computer resource access needs to be trusted. Access should be classified by the need for authentication, authorization, and finally trust. It is simplistic to say that all penetrations through a firewall, for example, need to be trusted. Procedures need to be developed and agreed upon to define the levels of required trust for each resource, and then a way for users to prove that these are satisfied.

If labs use enclaves to instantiate their levels of protection, means must be defined to compare these levels at different sites, and to translate between them if necessary.

In general, a person or organization owns resources, and it is that entity that must be responsible for granting access to the resource and for determining the conditions of such access. These conditions must then be harmonized with the policies of the lab to present an external security interface that is well defined and implementable.

6.3.4 Research Topics

Enclave Policy Engine

Creating, maintaining, and enforcing policy is difficult to do without some sort of graphic user interface and associated software to allow one to specify and to enforce a policy. As a simple example, the fields of all users' PKI certificates could be extracted and presented in a Web page form. These fields could be used in Boolean policy algebra to control access, say, by using a servlet engine on the Web servers (which is one of the functions of the Akenti [] system). In general, however, policies need to be more complicated than just file access restrictions. For example, they might only allow access for a period of time, or they might have to make a decision within an executable according to who is the executer. As another example, John

Barkley at the National Institute of Standards and Technology (NIST) has implemented a Web-based policy engine for role-based access control [20].

Hostile remote access

An enclave will probably need to allow access to its resources from outside of the enclave. For example, members of the enclave might need to access it from home, or customers might need to access a facility within the enclave. The labs do not control the computers used for such access, so in general, they might have been compromised, and should be assumed to be hostile. It is probably impossible to assure complete security of remote access, but there are two possible useful approaches to this problem.

In a manner similar to a virus scan program, or to the Norton Utilities, a program could be run on the remote computer and check whether there are up-to-date anti-virus definitions and whether a scan has been run recently. It could check for which ports were being used, and whether there was some sort of personal firewall, and so forth. If the machine passes the test, a PKI certificate could be issued that would be valid for a day that would certify this machine as a "clean" remote host.

A second approach might be to require that the remote host download and run access software provided by the enclave (each day, perhaps), for example a Java applet. This method would cut down on the chances that software used for access has been compromised and is launching attacks in the background.

Both of these approaches need to be investigated.

Protecting a Distributed Enclave

An enclave may not coincide with physical network boundaries, or even with a collection of laboratory-defined enclaves. Therefore, there is no single point of access to the enclave that can be monitored for intrusions. The lab-based protection systems are also probably not configured to enforce the fine-grained policies of CDs. For example, in the Diesel Combustion Collaboratory, information flow between the enclave sites must be constrained because of its proprietary nature, and it is up to the collaboratory to ensure that this is done. Concomitantly, the resources used to provide the enclave protection might come under attack in new ways that do not apply to the lab infrastructure as a whole.

Other Research Topics

- Define what components have an impact on a system, for example, user authentication, network services.
- Define the characteristics of each relevant resource, for example:
    - For user authentication, what Kerberos version number, user/password characteristics including length and time-out of passwords, PKI identity certificates including trusted CAs (Certificate Authorities).
    - What network services are allowed: ssh, rsh, ftp? Is anonymous ftp allowed, and how are passwords protected? Are Web servers allowed and, if so, what dynamic content types are allowed?

- What operating systems are allowed? What versions?
- What physical security is provided?

- Attempt to rank the characteristics of each resource.
- Equal rankings should be allowed. Ranking may depend on details of the component being ranked, e.g., enforced expiration of passwords may make one user/password scheme more secure than another. Whenever possible, define a rank in terms of objective and automated tests that a system can pass, such as known vulnerabilities for operating systems, or time it takes to crack passwords on a system.
- Pick sets of values to define an enclave level.
- Enclave definitions might be based on their intended purpose, e.g., to protect data and access to resources only from non-enclave members or to protect proprietary data within the enclave from non-authorized enclave members. Definitions might also be based on how "secure" an enclave is based on the values for a broad range of characteristics.
- Develop suites of tests to validate that a system is at a specified enclave level. These tests would probably be combinations of tests designed to measure the security level.

7 Code Safety

Barton Miller, University of Wisconsin
Mary Thompson, Lawrence Berkeley National Laboratory
Michael Fisk, Los Alamos National Laboratory

7.1 Mobile Code

7.1.1 Motivation

A characteristic of open scientific environments is the mobility of code throughout the system. The author or user of a piece of code is likely to be from a completely different organization than the owner of the machine on which the code runs. Mobile code comes in many varieties, affecting almost all aspects of a system:

- *Dynamic Scheduling Environments such as Condor or Globus:* A key to collaborative environments is sharing resources. Condor allows sharing of compute resources by running a computational task on any available computer (across organizational boundaries); programs move around as computing resources become available. Globus jobs are pre-loaded onto hosts. As computational tasks migrate, we must insure the safe execution of these tasks and the integrity of their data, as well as the safety of the remote hosting site.
- *Screensavers / Peer-to-Peer:* Similar to dynamic scheduling environments, a growing number of organization are trying to harness the idle cycles of desktop machines by executing remote code whenever the screensaver is running. These organizations vary from research-based, such as SETI@home, to commercial ventures, such as Entropia. Since these programs dynamically import foreign programs onto a desktop machine, the safety of the desktop environment is in question.

- *Collaborative Documents:* Email, documents, and spreadsheets are frequently exchanged in a collaborative work environment. Today, these documents transcend the limits of traditional static documents by including markup language and procedural code that is evaluated whenever the document is viewed. We must insure that these documents can be accepted without introducing threats such as viruses.
- *Java Applets and Java Agents:* Java is designed to allow the importing of applets into a Web browser, with minimal danger to the browser (and its host). But the browser might manipulate the applet and cause it to make malicious requests back to its originating server. Any server that originates applets is at risk. These applets are a commonly used mechanism in collaborative environments, being the tool of choice for user interfaces.
- *Rich Media and Multimedia:* In a collaborative environment, sharing can often include complex objects such as images, documents, video clips, sound clips, and 3D models. To view and manipulate these objects, we require media plugins for browsers. Each time we load a new plugin, we introduce an unvetted pieces of foreign code into our local computing environment.
- *Automated Updating Systems:* Many common software codes attempt to keep themselves up – to date by contacting their home server and loading new software as needed. Virus-protection packages update themselves in this way to keep abreast of new threats; Microsoft Internet Explorer 6 and the Windows 2000 operating systems install patches automatically using such updates; and America Online (AOL) software does this type of update on each connection. Each update delivers unvetted pieces of foreign code into the local computing environment.
- *Databases:* Large collections of scientific data are often stored in commercial database systems. Scientific data differs from commercial data in that it can be structurally complex, requiring database support from complex object types. Modern databases extend their type handling with dynamically loaded modules (e.g., the “Data blades” of Informix). Loading new modules into your database system can provide threats to the integrity of your data and the underlying operating system.

7.1.2 Issues

The common threat in the above examples is that programs and data move into, out of, and within collaborative environments. Both the safety of the computing platform accepting the foreign code and the safety of the code and data arriving on a foreign machine must be preserved. Accepting foreign code onto your computer requires methods of determining the safety of that code. The determination might be done statically, checking the safety of the code before execution. Alternatively, we can dynamically check the code safety, providing execution limits (“sandboxing”) and detecting when an inappropriate operation is attempted.

Once we have found an errant program, we are interested in understanding its path of entry, its behavior, the particular inappropriate activities, and the construction of the program. As part of this analysis, we are also interesting in understanding the goal of the program (e.g., what was it trying to access or change), what it was going to do after it effected its access or change (e.g., replicate, spoil data, steal data, deny services), and what would trigger the activity.

7.2 Reliability of Code

7.2.1 Motivation

The vast majority of computer intrusions are caused by bugs in network software that allow unauthorized access to the host system. Security research in the area of software reliability must be stimulated to change this trend and create more secure systems.

7.2.2 Issues

As computing environments become more distributed and there is greater use of active content and mobile code, almost every piece of software becomes involved in distributed communications and is prone to security vulnerabilities. Open scientific computing environments develop a significant amount of network application code. Grid computing, in particular, relies on large amounts of software developed by the scientific community rather than industry. Even when programmers have good intentions and are conscious of security issues, it is extremely difficult to write software that does not possess security-significant bugs. Commercial, off-the-shelf software (COTS) also possesses unacceptably large numbers of security-significant bugs.

Security research in the area of software reliability must be stimulated. Rather than relying purely on human software engineering processes, it is necessary to also change the programming and operating environments in order to generate more reliable code. Many of today's environments have "fail-open" behaviors in which software bugs allow additional access rather than preventing access. Automated processes may not detect high-level logic errors, but should be able to prevent implementation errors (such as bounds checking, tmp file races, variable argument functions, assumptions about the format of remote input, etc.).

The large base of existing software cannot be trivially discarded. It is necessary to secure applications that have been written in unsafe languages for unsecured operating environments. Further, the source code for many applications is unavailable and, as a consequence, the safety of binary executables must also be analyzed.

7.2.3 Research Issues

We divide the research issues relating to determining the safety of foreign code on our trusted systems into three categories: static, dynamic, and forensic. We must also consider techniques to protect our code and data, when it runs in a foreign (and untrusted) environment.

Static Techniques

- How do we specify safe behaviors?
- Safety analysis of codes written in unsafe languages such as C, given the source code.
- Generating verifiable proofs of program behavior. These proofs might originate from augmented compilers or from binary analysis tools.
- Given access to large amounts of computational power, are any new solutions feasible?
- How do we statically vet code in advance of its execution? What techniques from formal compiler static analysis can be used to determine if the code will behave within reasonable

limits? The goal is to analyze the program based on its binary (executable) format, without access to its source code.

- Can static techniques be used on large programs? Can static analyses scale? Can we build tools focused on particular threats (such as "stack smashing" or buffer overflow attacks), such that this more focused analysis will be more efficient?
- Given the computation demands of static analysis, can we use our available high-performance computing systems to extend our reach in these types of analyses?

Dynamic Techniques

These techniques also apply to problems in forensics:

- What techniques can we use to control the execution of foreign code, such that when it attempts to perform an inappropriate operation, we immediately detect it?
- What operating system facilities can we use, and what new ones need to be developed to detect all reasonable inappropriate behaviors?
- How can we apply "binary rewriting" to transform a foreign program into one that will dynamically detect an unsafe action?
- How can sandboxing techniques (operating environments that enforce the principle of least privilege) be applied to this problem domain?

Forensics

- How we design tools to isolate the specific inappropriate behavior attempted by the program?
- Can we determine the source of the program and what triggered its malicious behavior?

Safe Execution in a Hostile Environment

- How can we execute our code on a remote (and questionable host) and trust the answers returned by the program? How can we ensure that requests made by the remote program during its execution will not cause harm?
- How can we protect data and algorithms that are sent with the remote program from being exposed? Are there encryption or partitioning schemes that can help with this problem?
- Programming language safety for high-performance applications and services.
- Reusable, secure software components.

8 Towards a Cyber-Security Science

Thomas D. Ndousse, Office of Advanced Scientific Research, U. S. Dept. of Energy

Cyber-security faces the kind of crisis that software engineering faced a decade ago: (1) the lack of well-understood methodologies to specify, develop, and test secure systems, (2) the inability to specify and validate with a reasonable degree of certainty the degree to which a system has

been rendered secured, (3) the lack of a well-trained workforce to develop and maintain secure systems, and (4) the lack of tools and formal techniques to address the complex issues emerging in the field. Cyber-security, when viewed in the context of the global Internet, pervasive computing, and large-scale distributed scientific collaboration, exhibits many complex phenomenal properties of biological systems that defy many classical and ad hoc approaches currently being employed to understand and build secure, trusted, and reliable distributed systems. Experience has shown that many complex problems such as biology, chemistry, etc., are better understood and studied through the use of scientific methods – a body of systematic formalisms that enable researchers to discriminate between rival quantitative theories and select one that has the correct encoding of qualitative content. Extending the scientific method to cyber security would eliminate the current ad hoc approaches and provide the following contributions to the field:

- **Cyber-Security Science Discipline** – A scientific discipline within which the fundamental science of secure distributed systems can be formally explored, extended, and researched using well-understood scientific principles.
- **Cyber-Security Workforce** – As an established scientific discipline, cyber security will provide a mechanism to produce trained professionals in an academic setting to apply cyber-security methodology and principles to the complex issues of cyber-security.
- **Cyber-Security Metrics** – To enable the development of cyber-security tools to verify and validate systems against security objectives and requirements.
- **Cyber-Security Modeling and Analysis** – To enable the development of robust mathematical techniques, cyber-security performance bounds, and qualitative comparison of candidate cyber-security techniques and systems.
- **Secure Computational Complexity** – To address at the fundamental level cyber-security issues in related disciplines such as programming languages, computer organization and operating systems, software engineering, and network protocols design.
- **Trust modeling** – Language for expressing, validating, and modeling trust in cyber-space and in large-scale scientific collaborations.
- **Cyber-Forensic Science?**

Part IV: Conclusions

This fourth workshop in a series of DOE Office of Science – Defense Programs workshop focused on the security issues for open, scientific environments. These environments are the norm in the unclassified scientific R&D that is the mission of the Office of Science. They are typical of high energy physics, astronomy and astrophysics, accelerator based experiments in materials and life sciences, etc. In other words, these environments are the norm for modern science.

The workshop participants were drawn almost equally from the Science and Defense Programs DOE Labs, together with several universities and the DOE ASCR/MICS office. Participation by the Defense Programs Labs was very useful to the workshop, and because the open scientific environment has many of the same characteristics once inside the classified computer environment, it was also useful to the Defense Programs Labs.

The workshop examined the significance of open collaborations to DOE's mission, the cybersecurity issues in these environments, some aspects of the future computing environment, and cybersecurity threats. Following this, issues were identified where computer science R&D could contribute to increasing the security of open science environments.

Because of it's major scientific facilities and science mission, DOE must – and does – have a leadership role in building and using large-scale collaboratory environments. Therefore, DOE must take a leadership role in protecting these environments or they will not reach their potential for fostering new and highly productive ways of doing science.

The workshop examined example scenarios from half a dozen DOE science programs in order to characterize the open DOE science environment, and the security issues in those environments. From this examination we concluded:

1) Collaboratories are the combination of human collaborators, computer mediated services, and compute, data, and instrument resources drawn from all over the world that support the large-scale collaborations that are necessary to address the hard science problems that are at the core of DOE's Office of Science mission.
2) Change is the norm in this environment, not the exception: new computing and data services are continually being developed to meet new challenges and more effectively apply computing and data analysis to solve scientific problems – rapid prototyping of digital services is how this is done. A rich set of computer mediated services is critical for collaboratories: security cannot be obtained by exclusion of all but the few most common services.
3) Grid services providing access to resources used by scientific communities – uniform CPU access, resource discovery, resource management, uniform data archive access, security, etc. – will be the Internet Services for 21st Century science, and Collaboratories will be built using this infrastructure.

4) Security – denial of service, access control, confidentiality – is a major concern that must be addressed for viable Collaboratories, but it cannot impede the free flow of ideas and information, and access to computing resources.
5) Collaboration – sharing resources across administrative and security domains – will not happen without approaches to security that both protect and allow access.
6) A wide range of security R&D topics have been identified where DOE has the expertise to make a major contribution toward realizing collaboratories by defining and implementing appropriate security that protects open science environments AND allows widely distributed collaboration at the same time.

Notes and References

[1] "Worldwide Distributed Analysis for the Next Generations of HENP Experiments," H. Newman. In *Computing in High Energy and Nuclear Physics.* 2000. Padova, Italy. http://chep2000.pd.infn.it/abs/abs_e385.htm

[2] "Data-Intensive Computing," R. Moore, C. Baru, R. Marciano, A. Rajasekar and M. Wan, in *The Grid: Blueprint for a New Computing Infrastructure*, I. Foster and C. Kesselman, Editors. 1999, Morgan Kaufmann. p. 105-129.

[3] "The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Data Sets", A. Chervenak, I. Foster, C. Kesselman, C. Salisbury and S. Tuecke. J. Network and Computer Applications, 2001.

[4] "NASA's Information Power Grid," IPG. http://www.ipg.nasa.gov

[5] "Particle Physics Data Grid", PPDG. 2000. http://www.cacr.caltech.edu/ppdg/

[6] The Extensible Computational Chemistry Environment (Ecce) is a domain encompassing problem-solving environment for computational chemistry. "Extensible Computational Chemistry Environment (Ecce)," D. R. Jones, T. L. Keller, K. L. Schuchardt, H. L. Taylor and D. K. Gracio. 2001. http://www.emsl.pnl.gov:2080/docs/ecce/

[7] The Globus project is developing fundamental technologies needed to build computational grids. Grids are persistent environments that enable software applications to integrate instruments, displays, computational and information resources that are managed by diverse organizations in widespread locations. "The Globus Project," Globus Project. 2001. www.globus.org

[8] "Materials MicroCharacterization Collaboratory", N. J. Zaluzec, M. A. O'Keefe, M. T. Postek, D. E. Newbury, E. Kenik, E. Voelkl, M. C. Wright and J. Mabon. 1997. http://tpm.amc.anl.gov/MMC/

[9] The DOE Science Grid's major objective is to provide the advanced distributed computing infrastructure based on Grid middleware and tools to enable the degree of scalability in scientific computing necessary for DOE to accomplish its missions in science. "DOE Science Grid," Science_Grid. 2001. http://www-itg.lbl.gov/Grid/

[10] "Supernova Cosmology Project," S. Perlmutter and e. al. 2001. http://www.supernova.lbl.gov

[11] The "Cosmology Tutorial" is a good introduction to cosmology in general, and also specifically discusses supernova cosmology (at http://www.astro.ucla.edu/~wright/sne_cosmology.html) "Cosmology Tutorial," N. Wright. 2001, UCLA. http://www.astro.ucla.edu/~wright/cosmolog.htm

[12] NERSC is one of the largest unclassified scientific supercomputer centers in the US. It's mission is to accelerate the pace of scientific discovery in the DOE Office of Science community by providing high-performance computing, information, and communications services. NERSC is the principal provider of high performance computing services to Office of Science programs -- Magnetic Fusion Energy, High Energy and Nuclear Physics, Basic Energy Sciences, Biological and Environmental Research, and Advanced Scientific Computing Research. "National Energy Research Scientific Computing Center," NERSC. 2001. www.nersc.gov

[13] The Diesel Combustion Collaboratory will facilitate collaboration among the participants in a distributed research project, specifically the Heavy Duty Diesel Combustion CRADA (Cooperative Research and Development Agreement). Diesel_Combustion_Collaboratory. 2000. http://www-collab.ca.sandia.gov
[14] "Groove software lets you create secure shared spaces where you make instant and direct online connections with others to share information and get things done." Groove. www.groove.net
[15] "Computer Immune Systems," S. Forrest. http://www.cs.unm.edu/~immsec
[16] *The Grid: Blueprint for a New Computing Infrastructure*, I. Foster and C. Kesselman, eds. 1998, Morgan Kaufmann. http://www.mkp.com/books_catalog/1-55860-475-8.asp
[17] "Grids as Production Computing Environments: The Engineering Aspects of NASA's Information Power Grid," W. E. Johnston, D. Gannon and B. Nitzberg. In *Proc. 8th IEEE Symposium on High Performance Distributed Computing*. 1999: IEEE Press.
[18] "A CORBA-based Development Environment for Wrapping and Coupling Legacy Codes," G. Follen, C. Kim, I. Lopez, J. Sang and S. Townsend. In *Tenth IEEE International Symposium on High Performance Distributed Computing*. 2001. San Francisco.
[19] "Certificate-based Access Control for Widely Distributed Resources," M. Thompson, W. Johnston, S. Mudumbai, G. Hoo, K. Jackson and A. Essiari. In *Eighth Usenix Security Symposium*. 1999. http://www-itg.lbl.gov/Akenti/papers.html
[20] "Role Based Access Control," NIST. 2001. http://csrc.nist.gov/rbac/

Acknowledgements

This work was funded in part by the U.S. Dept. of Energy, Office of Science, Office of Advanced Scientific Computing Research, Mathematical, Information, and Computational Sciences Division (http://www.sc.doe.gov/production/octr/mics) under contract DE-AC03-76SF00098 with the University of California.

This document is Lawrence Berkeley National Laboratory report number LBNL-48862.